

Automatic Detection of Cyber Security Events from Turkish Twitter Stream and Newspaper Data

Özgür Ural¹^a and Cengiz Acartürk^{1,2}^b

¹*Informatics Institute, Cyber Security Graduate Program, Middle East Technical University, Ankara, Turkey*

²*Informatics Institute, Cognitive Science Graduate Program, Middle East Technical University, Ankara, Turkey*

Keywords: Cyber Security, Event Detection, Turkish, Twitter, Hürriyet Newspaper.

Abstract: Cybersecurity experts scan the internet and face security events that influence user and institutions. An information security analyst regularly examines sources to stay up to date on security events in the domain of expertise. This may lead to a heavy workload for the information analysts if they do not have proper tools for security event investigation. For example, an information analyst may want to stay aware of cybersecurity events, such as a DDoS (Distributed Denial of Service) attack on a government agency website. The earlier they detect and understand the threats, the longer the time remaining to alleviate the obstacle and to investigate the event. Therefore, information security analysts need to establish and keep situational awareness active about the security events and their likely effects. However, due to the large volume of information flow, it may be difficult for security analysts and researchers to detect and analyze security events timely. It is important to detect security events timely. This study aims at developing tools that are able to provide timely reports of security incidents. A recent challenge is that the internet community use different languages to share information. For instance, information about security events in Turkey is mostly shared on the internet in Turkish. The present study investigates automatic detection of security incidents in Turkish by processing data from Twitter and news media. It proposes an automatic prototype, Turkish-specific software system that can detect cybersecurity events in real time.

1 INTRODUCTION

1.1 Motivation and Objectives

Security awareness tools help security analysts to protect an institution's sensitive and mission-critical data from being stolen, damaged, or compromised by attackers. The duration between the disclosure of a new vulnerability and the moment when the security analyst becomes aware of it is crucial for taking appropriate countermeasures in a timely manner.

Twitter is a major source of up to date information. Twitter has 330 million monthly active users worldwide (Phan et al., 2020). Turkey is the fifth country in the list of leading countries with nearly 9 million active users, as of January 2019 (Okay et al., 2020). Twitter users can tweet in any languages they select. Although there are no statistics about the use of Turkish by Twitter users, it is very likely that most of the

Turkish Twitter users share their tweets in their native language.

A review of the literature and recent state of technology reveal that most of the research conducted on security event detection has been developed for analyzing text in English or other popular languages such as Portuguese language (Duarte et al., 2018) using Big Data (Seth et al., 2017). As of our knowledge, research is lacking on real-time security event detection in Turkish language streams. Given the significant share of the use of the Turkish language on the Internet, it is necessary to develop security event detection tools that process Turkish data. Internet usage penetration in Turkey is %72 with 59.36 million internet users, and active social media penetration in Turkey is %63 with 52 million people (Alan, 2020). With emerging internet adoption in Turkey, there are much timely information shared in Turkish. Recent event detection systems which developed for English texts are not useful for Turkish texts mining. Therefore, in order to use Turkish texts at detection of cybersecurity events, we should develop Turkish language-specific

^a  <https://orcid.org/0000-0003-1329-4303>

^b  <https://orcid.org/0000-0002-5443-6868>

methods and algorithms.

Social media is not the only option to extract information as such. A security analyst has a wide range of sources available such as the specialized press, blogs, forums, news agencies, newspapers, and so on to gather cyber threat information. Although, their initial source of information for detecting such security events is usually social networks. An alternative way to extract information about security events is newspapers. After the emergence of a trending event, users increasingly share posts about it on social media. For instance, a DDoS attack to a service or a website is usually recognized and reported by social media users first, and they share the information on online platforms, by posting tweets such as “X website is unreachable”.

An autonomous system which can use various data sources for security event detection has the potential to be beneficial for a security analyst. We designed and developed a software system capable of detecting and monitoring cybersecurity-related events over the Twitter Stream in Turkish. In its recent version, it can process several millions of documents per day and detect security events. To gain more accurate results, we added the Hürriyet Turkish newspaper stream to Twitter, for analyzing and detecting security events. The software solution’s infrastructure supports adding new data resources, thus providing flexibility. For example, it is possible to expand the system by adding LinkedIn, Facebook website streams to gain more complete and accurate results.

We designed the system as a framework to make it useable for further research. Turkish datasets are used in various research areas like text classification, author detection, automatic question answering. However, finding datasets in Turkish is difficult since there are limited accessible datasets online. By means of this research software framework, researchers will be able to access security event datasets in Turkish. Moreover, they will be able to select and modify their queries by changing keyword vectors, thus changing the content of information to be extracted from online sources. We validated the proposed approach using several detected events already shared in Turkish-in online platforms. By means of automatic event detection systems, a security analyst establishes situation awareness in cyberspace and take countermeasures against new threats. For example, a security analyst who is working for a Turkish institution may use local websites APIs like Eksisozluk API e-Devlet API or libraries/frameworks developed for focused Turkish people. If these API’s, libraries or frameworks have vulnerabilities, and someone discovers them, they are probably discussed and announced within social me-

dia like Twitter in Turkish. It is likely that Turkish newspapers publish it as breaking news too. To detect such events automatically, the software system must listen to Turkish data sources and process the text in Turkish. Our research aims at meeting these requirements by proposing a software system and framework for security event detection.

1.2 Routine Tasks of an Information Security Analyst

Information security analyst’s the primary responsibility is to take countermeasures for protecting organizational-level, mission-critical and sensitive information, as well as being prepared for cyber-attacks (Sohime et al., 2020). To be prepared for a cyber-attack, they use various tools and systems. One of their responsibilities is to analyze data and to recommend changes to managers. However, security analysts are not authorized to implement changes. Their main job is to keep cyber-attacks out.

In practice, a security analyst spends approximately one hour per a working day to get caught up on the latest security news through bulletins, forums, news, social networks and so on to identify new threats. They further spend two to three hours by repeated investigation of potential security incidents using online resources. They spend the rest of their daily time with manually copying and pasting information from disparate and siloed tools to correlate data. They generally face with ten to twenty challenges daily such as monitoring security access, analyzing security breaches to identify the root cause, verifying the security of third-party vendors and collaborating with them to meet security requirements and so on. (Sohime et al., 2020) Their investigation time gives cyber attackers advantages if it is long enough, and it is challenging for a security analyst to keep up with threats. A manual investigation of security events is not sustainable without automation. To make it sustainable, automated Natural Language Processing analysis tools and text mining methods need to be used.

1.3 Relevant Work

The identification of victims affected by cyber-attacks is a major subdomain of research in cybersecurity. One of the research field focuses on cybersecurity events detection using English text in Twitter. For example “Automatic Detection of Cyber Security Related Accounts on Online Social Networks: Twitter as an Example”. In that paper (Aslan et al., 2018), they use machine learning techniques; they investigated to

find a method of whether social media accounts related to cybersecurity. To prepare their dataset to use in their research, they develop a crawler with Twitter API using Python programming language. Another notable paper in this domain is "Processing tweets for cybersecurity threat awareness"(Alves et al., 2021). They tested a quantitative evaluation considering all tweets from 80 accounts over 8 months (more than 195,000 tweets), it shows that their approach timely and successfully finds most of the security-related tweets related to an example IT infrastructure (rate positive rate greater than 90 %), incorrectly selects a small number of tweets as relevant (false positive rate less than 10 %).

Another subdomain of research is event forecasting. The researchers try to estimate the DDoS attacks that have not yet taken place by processing Twitter data. They tried to obtain this information using six popular supervised classification models. To illustrate, one of the models which they used is the "negative term count". Neg-Term-count is the baseline sentiment-based model. They count the negative words from tweets each day, forecasting an attack if the number of negative words is more significant than a threshold, which is the average number of negative words on training data.

Another subdomain of research is Drive-by Download Attack Prediction. Cyber attackers may use the URL abbreviation method to show malicious websites as if a harmless website and share them on twitter as an abbreviated URL. Twitter users may believe in this deception and click on such website abbreviations, and these links can harm the users. "Prediction of Drive-by Download Attacks on Twitter" is an example which researches this field. (Javed et al., 2019) They have explored what we can do to prevent such malicious websites from being clicked like a safe website due to this kind of abbreviation. They try various methods such as detecting malicious software infection from the increase in the use of CPU or RAM with using Honeypot.

Another subdomain of research is cyberattack detection using social media. A sample study on this field is "SONAR: Automatic Detection of Cyber Security Events Over the Twitter Stream". They developed a self-learning framework called Sonar. (Petersen,) Sonar can automatically capture events related to cybersecurity by processing twitter data. Developers give the system some keywords to follow. The system can find other keywords to followed related to cybersecurity with the help of previously given keywords. They have also benefited from big data technologies. For the architectural design of our system, we use this research in our present re-

search. Another example is "Crowdsourcing Cybersecurity: Cyber Attack Detection using Social Media". (Khandpur et al., 2017) It is another study on detecting cybersecurity attacks by processing Twitter data.

2 SYSTEM ARCHITECTURE, DESIGN AND METHODOLOGY

In this chapter, we explain the software system's architecture and design and methodology. Firstly, we explain the general approach. Then we present data collection using Standard Twitter API, Twitter Premium API, Hurriyet API, and Selenium. After that we mention how we can preprocess and process the data. Then we present how we detect a cybersecurity event with using anomaly detection which is one of the machine learning techniques.

2.1 The Approach

Figure 1 presents a general overview of the architecture and design. First, we need real-time streaming data to process. In order to establish a Twitter stream connection, the software uses statically defined the configuration file values. To gather the data in real-time, we use Standard Twitter API. We create cybersecurity-related Turkish keyword vector with using Term Frequency - Inverse Term Frequency analysis of past security incidents. We use this keyword vector to gather useful Twitter stream and Hurriyet Newspaper stream for our research. We use the language filter feature of the Twitter API in order to fetch only the Turkish Tweets. Hurriyet is a Turkish newspaper, therefore we did not need a language filter for it. To establish the Hurriyet Newspaper stream connection, the software also uses the configuration file. The architecture of the software system is implemented considering new data sources may be wanted to add. Before writing the fetched data to the database, both fetched data of Hurriyet Newspaper and Twitter are formatted to a suitable form for writing database.

After the normalization step, we move forward to Named Entity Recognition step of our pipeline. In this state, we use the predefined string vector, which currently includes institution names, government organization name, and country names. These strings represent the potential victims of security events. After that step, the software counts the number of mentions of the potential victims with searching the predefined string vector elements in the normalized texts which are stored in the database. We add daily

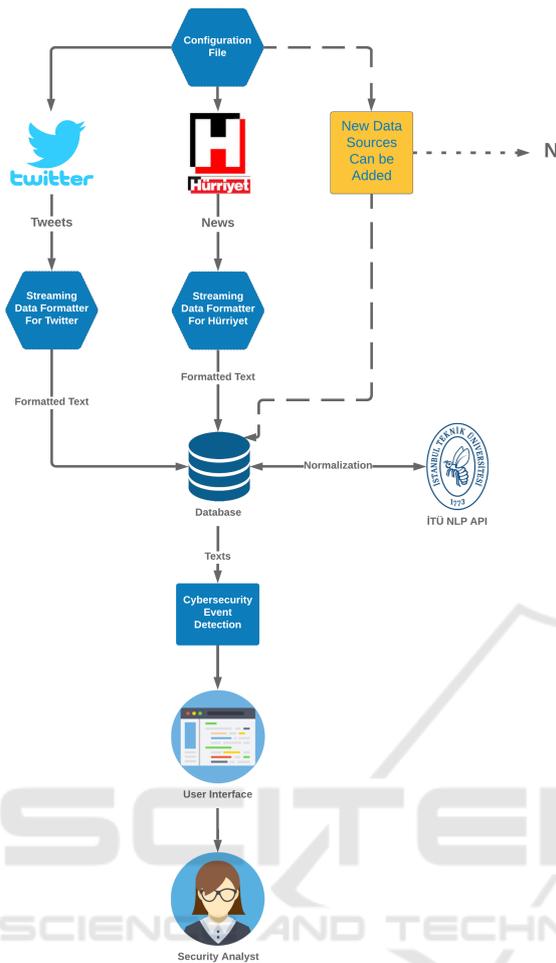


Figure 1: The General Overview of The System.

threshold values which is calculated dynamically. If the number of mentions is more than the thresholds value, we share this detected cybersecurity event within the user interface. The software repeatedly checks the database and analyzes new texts for detecting new cybersecurity events. If one of the possible victim's numbers of mentions in the cybersecurity-related database texts exceeds the threshold limit per day, the software system adds them to the table too. We show these detected events in a dynamically created HTML file. Security Analysts can see the detected security events from browser.

2.2 Selection of Cybersecurity Related Keywords Vector and Data Collection

To create an optimum version of cybersecurity-related keyword vector, we used term frequency-inverse document frequency(TF-IDF) technique, keyword-based

analysis, the statistical technique, and A/B testing. Even if the Tweets or the news are in Turkish language, there are widespread English cybersecurity terms used in Turkish texts. Therefore we create the vector using both English and Turkish keywords. It is a numerical statistic that intends to reflect importance a keyword or phrase is for within a document or a Web page in a corpus or in a collection. (Rajaraman et al., 2014) In order to identify our cybersecurity-related keywords vector, we used the Term frequency–Inverse document frequency technique. Firstly, we find one of the past important cybersecurity events related to Turkey from history. We select “nic.tr DDOS attack” as the cybersecurity event. Then we create three different training databases related to this attack with using Twitter Premium API. We select “nic.tr” as keyword and filter only the Turkish Tweets. With using tf-idf technique, we identify the most important words in these databases. Then we select the cybersecurity-related ones from the results of the tf-idf technique, and then add them to the cybersecurity-related keyword vector. The first query includes tweets containing nic.tr keyword at the dates between 10.12.2014 and 13.12.2015. These dates are the one year period of time before the nic.tr attack. Then we created another training database. We select the Tweets only at the day of the nic.tr attack on 14 December 2015. We analyze the tweets in the database with TF-IDF frequency analysis and do A/B test to select words from them to our cybersecurity-related keyword vector. Lastly, we created another training database. It includes the Tweets between 14 December 2015 and 28 December 2015. Within two weeks period of time, nearly 1000 Tweet had been tweeted related with “nic.tr”.

We analyzed their results and create cybersecurity-related keyword lists for each one of them. Then we used these keywords lists for A/B testing. The A/B test is a randomized experiment with two variants, A and B. It includes the application of the statistical hypothesis test or “two-sample hypothesis test” used in the field of statistics. The A/B test is a method of comparing two versions of the same variable and determining which of the two variants is more effective. (Fabritius, 2017) We compare the results of the A/B test and update the elements in the keyword vector according to their success rate. For A/B test we used the number of false-positive cybersecurity event detection and number of cybersecurity event detection. If a keyword significantly increases the number of false-positive detection, we do not add it to our cybersecurity-related keyword vector. On the other hand, if a keyword does not affect so much the false positive detection but increases the number

of detection we add it to our cybersecurity-related keyword vector list.

In order to collect data, we use Twitter and Hürriyet newspaper. Both Hürriyet API and Twitter API need seed keywords to query them. In order to collect Turkish stream data, we need Turkish cybersecurity terms. However, we cannot find a Turkish cybersecurity terms dictionary. Therefore, we research the Turkish cybersecurity terms and gather them as a list to use them in the query. Then we implemented a Python code to parse the Twitter website with using Selenium automation tool and chrome browser web driver and, created our desired training datasets. The selenium solution for fetching Twitter data is not a known method and it is firstly implemented by us.

Twitter is an online social networking service, which was created in October 2006 by Jack Dorsey, Even Williams, and Biz Stone. People use Twitter for various purposes. (Huberman et al., 2009) First of all, One of its usage examples is as a social messaging service. Users can interact with the other users, communicate with their friends and family, and share details of their lives. Secondly, users can use it as a microblogging service for sharing details of a person's life. Thirdly, users can use Twitter as a marketing tool for public relations. Many celebrities and politicians use Twitter for interacting with their audience. Lastly, Twitter is an information platform on which users can get news via broadcasting agents' or journalists' accounts fast and efficiently. Moreover, there are Twitter bots created by developers for a precise function like Bitcoin ticker bot will tweet every hour the price of Bitcoin in Turkish Lira. According to the first quantitative study on Twitter "What is Twitter, a Social Network or a News Media?" which is published in 2010 (Kwak et al., 2010), Twitter is more an information-sharing network than a social network. They found that result while working on Twitter follower graph. They decided that because of the low rate of reciprocated ties. People tend to use Twitter as a news feed by following multiple online news media, but other Twitter users will only follow "real" users. Twitter users can post a short message called tweet, which is limited to 280 characters, or retweet another user tweet. Photos, videos, or URLs can be added to the tweets. Users can follow other accounts and creates their networks. They can mention each other or reply to each other within their tweets. To identify what the tweet is about, users use word preceded by a hash sign (#). Twitter uses these hashtags to define trending topics, both locally and globally. Users use the trending topic lists to identify favorite subjects at that time on Twitter. In default settings, all Twitter accounts are public. Users can interact with each other

like replying other user's tweets, sending a private direct message, and so on. The Twitter API is a set of URLs. The URLs cant take parameters and let users access Twitter features like finding tweets which contain a set of specific words and so on. Twitter provides several APIs to get tweets. Twitter's Standart API allows users to get tweets which includes specific parameters. Moreover, the resulting stream can be filtered according to Tweet languages, geolocation and so on.

Our second data source, Hürriyet Newspaper. Hürriyet is one of the major Turkish newspapers, founded in 1948. As of January 2018, it had the highest circulation of any newspaper in Turkey at around 319,000. We can make 12,000 request per day in Hürriyet Newspaper API. Therefore, the keyword list is essential to get relevant data in the result streams. Hürriyet API is an interface which enables the usage of Hürriyet data programmatically in web, mobile, or desktop applications. Developers can access Hürriyet newspaper data via standard HTTP requests. The resultant set of results is in JSON format.

2.3 Data Processing

Before writing the streaming data to our database, we need to format the collected texts. Firstly, we should select the needed keys from JSON streams of Twitter API and Hürriyet API. For example, Hürriyet API requests return related news in a JSON which has "Title of the News" key. The key can be useful for representing the detected event. On the other hand, there are unrelated or unuseful data in the JSON too, so we filter them and do not write in our database. We filter the Twitter API stream's JSON keys too and select the useful and relevant keys too. In our database, we have a 'Status' column. When we first write the texts to our database, we set the text's status with '0'. '0' means that the text is not processed yet, and it is raw data. We sent the raw data to ITU NLP API to normalize it. After the normalization step, we update the text with normalized text and update the Status column of the row which has the text with "1". After the row is processed to detect cybersecurity events, the Status column is set with "2". "2" means that the data processed before and there is nothing to do with that row of the table.

In the present research, we used a few Natural Language Processing techniques and Istanbul Technical University's Natural Language Processing API (Eryigit, 2014) for normalization of the texts. In order to develop automated systems, Natural Language Processing is one of the actively used concepts in text mining. It uses Natural Language Processing to de-

liver the system in the information extraction phase as an input. (Tan et al., 1999)

Turkish Natural Language Processing Tools and APIs developed by the Natural Language Processing group at Istanbul Technical University are available at “tools.nlp.itu.edu.tr” website. To be able to use the API, we used access token and an account for the token upon permission. The platform operates as a Software as a Service and provides the researchers and the students the state of the art NLP tools in many layers: preprocessing, morphology, syntax, and entity recognition. It is a web API, developers can access it with an HTTP request and can use GET or post method.

Text mining consists of a broad variety of methods and technologies. (Gaikwad et al., 2014) In this research, we used Keyword-based technologies and statistics technologies. Keyword-based technologies use the input based on a selection of keywords in text that are filtered as a series of character strings, not words nor concepts. (Wu et al., 2006) Statistics technologies leverage a training set of documents used as a model to manage and categorize text. In this research, we used keyword-based analysis and statistical techniques. We use two keyword vectors for keyword-based analysis. One of the keyword vectors stores possible victims who are tracked by our software solution. The other keyword vector stores the possible useful cybersecurity-related Turkish terms such as “hacklendi” and “erişilemiyor”. We analyze the results by comparing the past frequency statistics and current results as described in the Approach section. The text required for text mining for cybersecurity event detection purposes is gathered from online platforms.

From the previous steps of the software system, we get the possible cybersecurity-related texts from different sources. Then preprocess and process them and store them in our database. In order to detect the events and find the possible victim of those events, we prepared a named entity vector. This vector includes possible victims which we want to track. Currently, this list includes institution names, government organization names, and country names. The vector can be updated from changing the configuration file to change tracked entities. Then with using term frequency-inverse document frequency (TF - IDF) technique, keyword-based analysis, the statistical technique, and A/B testing; we analyze past cybersecurity events and create cybersecurity-related keywords vector.

As we explained in Approach section, we analyze real-time Turkish text data to detect cybersecurity events. In order to do this, we send requests to Twitter and Hurriyet newspaper with our cybersecurity-

related keywords vector and we add Turkish language filter to our request. The possible victim vector of the solution periodically checked in the database in terms of the number of occurrences. If the number of occurrences of a victim shows anomaly¹ according to its historical values, our solution detects them as a potential cybersecurity event and shows that events in the user interface portal.

3 IMPLEMENTATION

3.1 Multi-process Architecture

We use multi-processed system architecture in the implementation of the project. There are four processes as described in the subchapters below. These are Twitter API Stream to Database, Hurriyet API Stream to Database, ITU NLP API Normalization and Security Events Web Portal Processes. Twitter API Stream to Database Process continually gathers Twitter API stream. Then preprocess the data and write them to the database. Hurriyet API Stream to Database Process continually gathers Hurriyet API stream. Then preprocess the gathered data and write them to the database. ITU NLP API Normalization Process continually checks the database. If the process can find columns with status 0, then sent the columns to ITU NLP API servers to normalize them. After the normalization, the process writes back the texts to the database and update their status row with “1”. Security Events Web Portal Process continually checks the database to find columns with status row set with “1”. If it can find, it processes them to add the HTML page which security analysts can monitor the events from that page.

3.2 Microservice Architecture

Microservices are small, and independent services focus on doing a task at a time and ability to work together. Because the project has the potential to grow, we design it with following microservice architecture. With this design, our software became resilient. Failure in one service does not impact the other services of our project. For example, assume that ITU NLP API service stops to work for a while and does not respond to our project’s requests. Due to the microservice architecture of our software, the other services can continue to work even if our software has monolithic or bulky service errors in one service. Hurriyet API can still gather the streaming data, preprocess them, and write them to the database; Twitter API can still gather the streaming data, preprocess them,

and write them to the database and so on. Moreover, it has scalability. For example, if our database technology becomes insufficient for our software, we can change the database technology with a more suitable one. Furthermore, our software has less dependency and easy to modify its code and test them. Our software can understand by other developers since the processes represent the small piece of functionality. It is vital because our software solution will be an open-source project and will be used by other developers and researchers. Lastly, this architecture method gives us the freedom to choose technology. We can choose the best-suited technology for each of functionalities.

3.3 User Interface of the System

It is a simple dynamically generated HTML page which will be used by security analysts as a portal page of the system. A process continuously checks the database per minute to detect new data and use them to show the new cybersecurity events in this user interface.

4 RESULTS

In this chapter, we discuss the results of the cybersecurity events which are discovered by our software solution. We focus on what our software system succeeded and what it did not achieve. We share successful cybersecurity event detection samples and share the not successful cybersecurity event detection samples. As described in the previous subsection, it is a dynamically created HTML page. We divide the events by their dates. As cybersecurity event information, we represent an entity, a representative news title or tweet and a count which shows how many times the entity is seen in the data on the same day.

4.1 Historical Cybersecurity Event Detection Test with an Independent Dataset: Nic.tr DDOS Attack

To reach the best version of our software solution, we train our software with training data. In order to do that, we select an important cybersecurity event test that can our solution detect that cybersecurity event. Turkish Internet hit with massive DDoS attack started on 14.12.2015 and continues about two weeks long. Turkey's official domain name servers (Nic.tr) have been under a Distributed Denial of Service (DDoS) attack. We created 3 separate databases using existing

keywords. 2310 tweets were found when we pulled the tweets during the 1-year period before the attack. Then we analyzed these data, our solution can successfully find the cybersecurity events that took place for a year.

28 tweets were found when we pulled the tweets at the start day of the nic.tr DDOS attack. Results of this day data were important for us because we wanted to see that our solution could detect the event just after the attack happened. Then we analyzed these data, our solution can successfully detect the nic.tr attack as you can see in Figure 2.

Entity	Representative News Title or Tweet	Count
nic.tr	Biraz önce ".tr" uzantılı web sitelerine girişte sorun yaşandı. ODTU'nün (nic.tr) DNS sunucularına DDoS saldırısı yapıldığı söyleniyor.	3
ant	Biraz önce ".tr" uzantılı web sitelerine girişte sorun yaşandı. ODTU'nün (nic.tr) DNS sunucularına DDoS saldırısı yapıldığı söyleniyor.	1
suriyeli	KAÇIRIP FIDYE İSTEDİLER http://www.edessatv.com/sanliurfa/kacirip-fidye-istediler-h15535.html ... #sanliurfa #fidye #Suriyeli	2
çin	#istanbul'da #Fidye için kaçırılan #İranlı: "Siz olmasanız ölecektim" http://fb.me/3xAle924U	1
iran	#istanbul'da #Fidye için kaçırılan #İranlı: "Siz olmasanız ölecektim" http://fb.me/3xAle924U	1

Figure 2: Nic.tr Attack Start Day Detected Security Events Samples.

The nic.tr attack lasted for about two weeks. Therefore, we analyze that two weeks period (14.12.2015 – 28.12.2015) and we expected to detect the nic.tr attack. About 400 tweets were found when we pulled the tweets for the given period. After running our software solution with that database, the results were satisfactory. Our solution successfully detected the nic.tr attack as you can see in Figure 3.

As we explained before, we used one of the past cybersecurity incidents. We used Term Frequency - Inverse Document Frequency (TF-IDF) analysis of the news and tweets just before the cybersecurity event (premise) and immediately after the event. For immediately after phase, we used two different time intervals for testing. First one is the attack day, and the second one is the two weeks period after the attack. We used the attack day for sensitivity test. Our solution is accepted as successful in terms of sensitivity if it can detect the cybersecurity event at the attack day. We used two weeks of period after attack for certainty. Our solution is accepted as successful in terms of certainty if there is not so many (more than %30) false-positive cybersecurity event detection within two weeks period after a cybersecurity event. According to these success criteria, we train our software solution with the datasets and cybersecurity-related keyword lists. Then update our keyword lists according to the results. With using these lists, we

2015-12-14		
Entity	Representative News Title or Tweet	Count
nic.tr	Ulusal Siber Olaylara Müdahale Merkezi (@TRCert) Nic.tr kesintisinin DDOS saldırısı sonucu meydana geldiğini belirtti. #nictr	3

2015-12-16		
Entity	Representative News Title or Tweet	Count
rusya	ODTÜ/com.tr alan adı veren NIC.TR dDOS saldırı altında. Siber saldırı Rusya/İran kaynaklı olabilir.	3
iran	ODTÜ/com.tr alan adı veren NIC.TR dDOS saldırı altında. Siber saldırı Rusya/İran kaynaklı olabilir.	4

2015-12-26		
Entity	Representative News Title or Tweet	Count
ekonomi bakanlığı	Türk Hacker grubu Ayyıldız Tim, aralarında Rus Ekonomi Bakanlığı'nın sitesinin de bulunduğu 19 bin Rus sitesini hackledi.	10

2015-12-17		
Entity	Representative News Title or Tweet	Count
ant	Ülkenin Yurtdışından erişimi siber saldırı sebebiyle engellenmiş durumda, tr uzantıları iptal. Haber bile değil.	14

2015-12-15		
Entity	Representative News Title or Tweet	Count
türkiye	Dünya Türkiye'ni yeni yaptığı yazılımı konuşuyor nasıl sinyaller kesildi ve musula gece asker sevk edildi. Yazılım konusunda şaşkınlık...	53

Figure 3: Detected Security Events Samples between 14 and 28 December 2015.

tested the method and its accuracy in independent data set which is nic.tr DDOS attack dataset in the present section. As can be seen in Figure 2 and Figure 3, our software solution can successfully detect the nic.tr DDOS attack in terms of sensitivity and certainty and passed our test.

4.2 Successful Cybersecurity Event Detection Samples

In the following subsections, we share successful cybersecurity event detection samples and briefly try to explain how a security analyst can use this information.

4.2.1 WhatsApp Spyware Attack

As can be seen in the Figure 4, our software system can detect this event on 5 May 2019. However, there are two different entities about the same event.

Assume that a security analyst wants to track security events related to countries. When the security analyst sees the “WhatsApp Spyware Attack” event in

2019-05-14		
Entity	Representative News Title or Tweet	Count
meksika	WhatsApp 'casus yazılımı' hakkında neler biliniyor?	4
israil	WhatsApp, bir grup 'seçilmiş' kullanıcısının casus yazılımla hedef alındığını duyurdu	6

Figure 4: WhatsApp Spyware Attack Detection.

the user interface page with a country name entity, he should check the news or tweets to control whether it is a positive or false positive event detection. If it is a positive and useful cybersecurity event detection, the security analyst takes the required actions. There are two entities as “meksika” which is the Turkish synonym of Mexico, and “israil” which is the Turkish synonym of Israel. When we control the related news and tweets, we can see that an Israel firm named NSO Group performs the cyber-attack. Therefore “israil” is passing six times in the detected news and tweets. A Mexican journalist is affected by the cyber-attack. That is why we capture the “meksika” entity. The security analyst can notice such attack with following our software solutions user interface and can learn what the new WhatsApp cyberattack is, how one can protect from such attacks and so on from the related news and tweets.

4.2.2 Vulnerabilities in Remote Patient Tracking System Applications

STM is a Turkish software company which does researches about cybersecurity domain. They find a vulnerability about Remote Patient Tracking System Applications and share this information from Twitter and with using newspapers. Our software could detect the security incident which is happened on 26.04.2019 about “STM Warns about Remote Patient Tracking System Applications” successfully. If our software solution were to have used English texts as a data source, we could not detect such a cybersecurity event published in Turkish. Because of our software solution can analyze Turkish texts, we can detect such a cybersecurity event. This is an excellent example to show what our solution can do while the other solutions in the literature cannot do.

4.3 Unsuccessful Cybersecurity Event Detection Samples

Sometimes our software solution can detect false-positive events, or even it is a cybersecurity event, the detection may not be a useful event for security analysts. The following subsections examine such scenarios.

4.3.1 Sample False Positive Cybersecurity Event Detection

A sample not useful cybersecurity event detection detected by our software is like "Ömer bey inanamıyorum, gerçekten bunları siz mi söylüyorsunuz yoksa hesabınız mı hacklendi?". Even the tweet has "hacklendi" word, which is one of our keywords from our keyword vector; the event is not a real cybersecurity event. Analyzing such tweets to realize that it is not a real security event is hard for an automated system.

4.3.2 Sample Not Useful Cybersecurity Event Detection

Sometimes, even the detected event is a cybersecurity event; it may be a personal status primarily if it is published on Twitter. Security analysts should read the detected event from the user interface and decide that it is useful or not for her/him. Even if the detected event is not a personal cybersecurity event, the detected event may not be useful for security events. For example, an event may occur months ago, but a Twitter user or a Twitter bot may share the event in a Tweet as if it occurred newly. The time frame is configurable in our software system. Security analysts should configure the software detection timeframe according to their needs. For example, if a security analyst works for a big cybersecurity technology company and he/she wants to know more detected security events, he/she can set the timeframe longer. However, if another security analyst wants to know only the latest security events, he/she should set smaller timeframe in our software solution.

4.4 Evaluation of the Results

When we run our software with too much the cybersecurity-related seed keywords vector, our software system might receive more tweets than it can handle. Only about %20 of Twitter users are posting informative messages (Kral and Rajtmajer, 2017). Moreover, the false-positive cybersecurity event detection may significantly increase. It decreases the certainty of our software solution. On the other hand, if we run our software with too few cybersecurity-related seed keywords, our software system might not detect some cybersecurity events as fast as we expect from our software. It decreases the sensitivity. We expect that we can detect an attack on the day of the attack.

Although we can verify with other sources that the detected events are indeed occurring, or occurred, being sure that we have missed any events is very diffi-

cult. During our tests, we realized that we could miss small events. However, our solution does not miss any serious attack as far as we know. Sometimes our solution detects an already detected event as if it is a new cybersecurity event. Because our software uses one day as a period for its frequency calculation. For each day, all calculations start from zero again.

For a limited time, we run our software for testing purposes. At a sample test run of our software solution, our database of the software includes 437 entries. 186 of them is Twitter Tweets, and 251 of them is from Hürriyet Newspaper. After analyzing the entries in our database, our software solution can detect 29 cybersecurity events. 22 of them are positive detection, and 7 of them are false positive detection. Our software solution's success rate is approximately %76.15 These statistics show that this methodology works in the detection of cybersecurity events from Turkish texts with an acceptable success rate in term of certainty and sensitivity. Cybersecurity analysts can use our software with preparing our cybersecurity-related keyword vector and named entity vector and selecting a suitable time frame. Moreover, they can modify the keyword vector or named entity vector as they wish. If we add new data sources in the future, our software can work with bigger datasets and this leads to more accurate detection and it may increase the success rate percent of our software solution in terms of certainty and sensitivity.

5 CONCLUSION AND FUTURE WORK

5.1 Conclusion

In the last few decades, automation has been increasingly used in various field of people's life due to its benefits like cost reduction, productivity, availability, reliability, and performance. Cybersecurity is one of the fields which automation is often used. However, every automation software system has unique requirements to achieve its purposes. It leads to lots of research areas and unique automation systems. Automatic event detection is one of these research fields. Social media is one of the fastest ways to detect cybersecurity events because people and bots share such events in there. Newspapers are also shared such cybersecurity events and processing the newspaper data is relatively more straightforward because false-positive cybersecurity events are rarely shared in the newspaper websites.

In this research, we investigated automatic event detection of cybersecurity events from Turkish Twit-

ter Stream and Turkish newspaper data. We work on real-time data to achieve that our research can be used by security analysts. Existing publications about real-time cybersecurity event detection system generally use English texts to analyze and detect the events. We cannot find any research which use Turkish data sources to detect cybersecurity events. Using Turkish data sources for cybersecurity event detection is a new topic for literature. We believe that this research contributes to the literature by filling an uninvestigated field. We proposed an automated software system which works using different data sources, named entities, text mining methods, and "state of art" software techniques. Then we analyze the results of our software system. Even if our software system detects few false-positive cybersecurity events, it was often able to detect a useful cybersecurity event. For example, our software system can detect cybersecurity events such as WhatsApp Spyware, MuddyWater Attack, the Remote Patient Tracking System Applications vulnerability, Pirate Matryoshka Virus, Zombie Cookies threat. We concluded that event detection with using Turkish texts is applicable, and security analysts can use such a system like our software system as a helper tool.

5.2 Limitations and Future Work

Currently, our software system works on a local computer. When we move the software to a server (i.e. AWS), our software can work 7x24, which will be useful for detection success. If our software can work with bigger data, it will detect more events with more accurate event detection. To increase the streaming data, we are planning to add new Turkish data sources from other websites like Eksisozluk, LinkedIn, Facebook, and so on. This improvement will make our datasets an excellent resource for future work. After these improvements, our datasets can be useful not only for us but also the other researchers work on cybersecurity, cognitive science or computer science field. We shared our software solution as an open source project via Github under Apache-2.0 license and it can be reachable from "https://github.com/ozzgural/MSThesis" link. We are also planning to share our future works on there and according to users feedback, we are planning to refine our software tool. The developed scenario may be applied to the other languages with necessary modifications and this work is also in our future plans. Moreover, we do not handle the named entity recognition ambiguities yet. We are planning to handle them in the future.

REFERENCES

- Alan, G. A. E. (2020). The importance of marketing public relations for "new" consumers. *New Communication Approaches in the Digitalized World*, page 157.
- Alves, F., Bettini, A., Ferreira, P. M., and Bessani, A. (2021). Processing tweets for cybersecurity threat awareness. *Information Systems*, 95:101586.
- Aslan, c. B., Sağlam, R. B., and Li, S. (2018). Automatic detection of cyber security related accounts on online social networks: Twitter as an example. In *Proceedings of the 9th International Conference on Social Media and Society*, SMSociety '18, page 236–240, New York, NY, USA. Association for Computing Machinery.
- Duarte, F., Pereira, O., and Aguiar, R. (2018). Discovery of newsworthy events in twitter. pages 244–252.
- Eryiğit, G. (2014). ITU Turkish NLP web service. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Gothenburg, Sweden. Association for Computational Linguistics.
- Fabritius, M. (2017). How to motivate colouring app users.
- Gaikwad, S. V., Chaugule, A., and Patil, P. (2014). Text mining methods and techniques. *International Journal of Computer Applications*, 85(17).
- Huberman, B., Romero, D., and Wu, F. (2009). Social networks that matter: Twitter under the microscope. *First Monday*, 14.
- Javed, A., Burnap, P., and Rana, O. (2019). Prediction of drive-by download attacks on twitter. *Information Processing & Management*, 56(3):1133 – 1145.
- Khandpur, R. P., Ji, T., Jan, S., Wang, G., Lu, C.-T., and Ramakrishnan, N. (2017). Crowdsourcing cybersecurity: Cyber attack detection using social media. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1049–1057, New York, NY, USA. Association for Computing Machinery.
- Král, P. and Rajtmajer, V. (2017). Real-time data harvesting method for czech twitter. pages 259–265.
- Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 591–600, New York, NY, USA. Association for Computing Machinery.
- Okay, A., Gole, P. A., and Okay, A. (2020). Turkish and slovenian health ministries' use of twitter: a comparative analysis. *Corporate Communications: An International Journal*.
- Petersen, J. K. *Handbook of surveillance technologies*. CRC Press., Boca Raton, Fla., 3rd edition.
- Phan, H. T., Tran, V. C., Nguyen, N. T., and Hwang, D. (2020). Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model. *IEEE Access*, 8:14630–14641.
- Rajaraman, A., Leskovec, J., and Ullman, J. (2014). *Mining of Massive Datasets*.

- Seth, A., Nayak, S., Mothe, J., and Jadhay, S. (2017). News dissemination on twitter and conventional news channels. pages 43–52.
- Sohime, F. H., Ramli, R., Rahim, F. A., and Bakar, A. A. (2020). Exploration study of skillsets needed in cyber security field. In *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*, pages 68–72.
- Tan, A.-H. et al. (1999). Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge discovery from advanced databases*, volume 8, pages 65–70. Citeseer.
- Wu, S.-T., Li, Y., and Xu, Y. (2006). Deploying approaches for pattern refinement in text mining. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 1157–1161. IEEE.

