IEEE ANTS 2019 1570586094

Analyzing Malicious URLs using a Threat Intelligence System

Sampashree Nayak, Deepak Nadig and Byrav Ramamurthy
Department of Computer Science and Engineering,
University of Nebraska-Lincoln, Lincoln, NE, 68588 USA
Email: {snayak, deepaknadig, byrav}@cse.unl.edu.

***Abstract*—Threat intelligence and management systems form a vital component of an organization's cybersecurity infrastructure. Threat intelligence, when used with active monitoring of network traffic, can be critical to ensure reliable data communication between endpoints. Threat intelligence systems are well suited for analyzing anomalous behaviors in network traffic and can be employed to assist organizations in identifying and successfully responding to cyber-attacks. In this paper, we present a machine learning approach for clustering malicious uniform resource locators (URLs). We focus on a URL dataset gathered from a threat intelligence feeds framework. We implement a k-means clustering solution for grouping malicious URLs obtained from open source threat intelligence feeds. We demonstrate the effectiveness of our unsupervised learning technique to discover the hidden structures in the malicious URL dataset. Our URL keyword/text clustering solution provides valuable insights about the malicious URLs and aids network operators in policy decisions to mitigate cyber-attacks. The clusters obtained using our approach has a silhouette coefficient of 0.383 for a dataset containing over 11,000 malicious URLs. Lastly, we develop a probabilistic scoring model to calculate the percentage of malicious keywords present in a given URL. After analyzing over 72,000 malicious keywords, our model successfully identifies over 80% of the URLs in a test dataset as malicious.**

***Index Terms*—URL analysis, threat intelligence feeds, k-means clustering, machine learning.**

I. Introduction

Increasingly, organizations face cybersecurity challenges due to an ever-increasing number of persistent threat activities, irrelevant data flooding, and false positives. These attacks are forcing organizations to take better measures to control and prevent network attacks as a part of their cybersecurity policies. Cybersecurity threats are rapidly evolving to stay ahead of defensive security applications. Therefore, in addition to security applications, each organization requires a threat intelligence system to augment their work for better security of cyberinfrastructure. A threat intelligence system is a critical security tool that uses security intelligence to detect malicious activity inside the network. It works with all security functions across organizations to add context about malicious activities in the network. Threat intelligence systems do not form a separate security domain. Instead, they provide evidence-based intelligence to help make informed decisions about the security application in cyber-infrastructures.

Threat Intelligence plays a vital role in security analysis for any organization [1]. The threat intelligence system collects raw data about existing threats from a variety of sources that are then processed to provide intelligence feeds with the threat information. Automated security management solutions use the threat information to handle security tasks. However, to give a meaningful context to security operations such as anomaly detection, incident response, malware management, etc., threat intelligence data must be analyzed and processed to understand underlying patterns. This analysis can aid in detecting new vulnerabilities and attacks that are similar to existing threats. Network intrusion detection systems (NIDS) cannot detect new attacks and are therefore not suitable for identifying malicious behavior in a large cyber-infrastructures.

In this paper, we focus on analyzing a malicious URL dataset gathered from a threat intelligence framework. We implement a machine-learning approach to reduce the dependence on NIDS and to detect actual malicious activities in the network traffic based on known URL patterns. In this paper, we design a machine learning solution to detect malicious behavior in threat data intelligently. Our solution applies to real-world problems such as reducing false positive alarms in the data center. We analyze up-to-date threat information from well-known network intrusion detection systems and threat intelligence systems. We use an unsupervised learning technique to analyze malicious URLs.

First, we convert the text data to vectors using Term frequency-inverse document frequency (TF-IDF) [2]. We then use the k-means [3] clustering algorithm for URL keyword clustering and determine the optimal number of clusters using the elbow method. Next, we evaluate the quality of the clusters using silhouette analysis. Through our analysis, we identify if there are any commonalities between keywords in each cluster with a test dataset of 500 URLs. We find the top terms in each cluster using the TF-IDF score across the clusters, and design a probabilistic model by comparing the resulting keyword with a test set to identify malicious behavior.

The paper is organized as follows: In Section II, we present the related work; Section III provides a detailed discussion of our malicious URL feed sources and the dataset used by our proposed machine learning solution; In Section IV, we present our solution approach, architecture, data processing details, and our probabilistic scoring model; In Section V, we evaluate our clustering model and present the preliminary performance results; Lastly, in Section VI, we conclude our work.

II. Related Work

Recently, there is growing interest among researchers on threat intelligence systems [4], [5], [6] for enhancing security platforms. Numerous threat manangement frameworks have been developed for analyzing and managing cyber threats and attack information. Collaborative research into threats (CRITs) [7] is an example of a web-based tool that combines data analytics with a threat repository to manage threat data information. The tool serves as a repository for malware and attack data, and additionally, it provides researchers a platform for conducting malware analysis, correlating malware, and for targeting the threat data. The authors in [8] propose an automated threat management framework called Automated Threat Intelligence fuSion framework (ATIS). ATIS performs data collection and analysis. The ATIS controller serves as an interface between the data and application planes. ATIS correlates threat events by connecting different threat sources with new cyber threat events. HuMa [9], proposed a multi-layer model for investigation of complex security incidents such as Advanced Persistent Threats (APTs) in log files. HuMa processes large log files; heterogeneous information systems produce large amounts of data making it challenging to ensure end-to-end data protection. Large amounts of generated logs make security operations complicated and difficult. To tackle this challenge, HuMa proposes a multi-layer framework for the analysis of complex security threats. The authors in [10] present OSINET, a cyber threat inspection framework that uses open-source intelligence to enhance the security of critical cyber-infrastructure. OSINET inspects threats in critical infrastructure networks that are primarily disconnected from the public internet and enhances efficiency and management by subjecting it to cyber threat inspection. The work in [11] presents a survey of machine learning techniques in system security and defense. It also presents security issues associated with machine learning systems. Different from the above, our work focuses on the analysis of malicious URLs and on developing a clustering solution for accurately detecting malicious URLs based on keyword analysis.

III. Malicious URL Feeds and Dataset

Threat intelligence feeds are the collection of real-time streams of data that provide information on potential cyber-attacks and associated risks. Using these feeds, network operators have insights into possible sources of cyber-attacks through continuously updated information. The feeds provide threat intelligence information gathered from a wide variety of sources such as indicators of compromise, open-source feeds, organizational intelligence-gathering efforts, and shared information between organizations. Threat intelligence feeds contain suspicious domains, list of known malware hashes, IP addresses associated with malicious activity, threat signatures, etc. For feeds to be actionable, they have to be integrated into security applications so that threat information can be correlated with internal application traffic data like firewall and DNS logs. This allows network administrators to identify and mitigate potential cyber-attacks.

A. Dataset

We collect threat data from multiple threat intelligence systems. Threat data consists of malicious URLs, threat signatures, IP address, and network port information. For this work, we focus on analyzing malicious URLs. The malicious URLs in our dataset are collected from URLhaus feeds (available at https://urlhaus.abuse.ch/). URLhaus is a threat URL sharing project from abuse.ch. The project publishes malicious URLs that are routinely used for malware distribution. Our dataset, obtained from the above source, comprises of over 200,000 malicious URLs tracked on URLhaus. This database is updated periodically. We use a representative sample of 12,000 URLs for data analysis in this work. The dataset is partitioned into training and testing set, with 11,500 and 500 URLs, respectively.

IV. Solution Approach

Threat intelligence feeds from diverse intelligence sources contain false positives and are not always malicious. Instead, they are a mixture of malicious and non-malicious data. Our goal is to design a model to analyze malicious URLs based on their constituent keywords. Our dataset does not contain labeled information about the veracity of the URLs, and thus, we use an unsupervised learning approach to obtain useful insights about the dataset. We perform keyword analysis and text-based clustering on the malicious URLs using the k-means clustering. To obtain the clusters, we vectorize the data and evaluate the clustering performance.

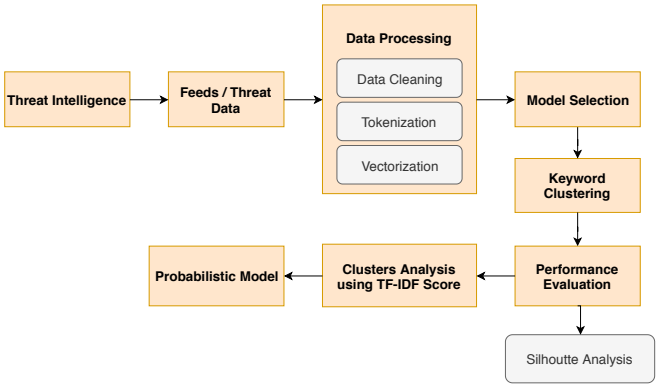

Fig. 1. Solution Architecture.

A. Architecture

Our proposed solution architecture is shown in Figure 1. We begin by processing the URL data gathered from threat intelligence feeds. We sanitize and tokenize the dataset, and then vectorize the dataset using term-frequency inverse document frequency (TF-IDF). The dataset is subject to k-means clustering and we use the elbow method to determine the appropriate number of clusters. From each cluster, we obtain the top terms and use these terms to create a probabilistic model for evaluating the model's performance. Clustering performance is also evaluated using silhouette analysis.

B. Data Processing

Data processing is an important step to convert unstructured data into a structured format. Our data processing approach has three steps: data acquisition, data cleaning, and preparation for the machine learning system. We gather raw data from the sources described in Section III-A. To analyze large amounts of text data in URL keywords and to uncover underlying patterns, text data cleaning and processing is necessary. We parse the malicious URLs for all delimiters such as '/' and '?' symbols. We remove all delimiters from our dataset to convert each URL into an array of keywords. After processing 11,500 malicious URLs, we obtain a dataset consisting of over 72,000 keywords. We convert any IP addresses in the dataset to their corresponding domain names.

In this work, term frequency (TF) represents the frequency of keywords in each URL. It is the ratio of the number of times the keyword occurs in a URL to the total number of keywords in that URL. The term-frequency is given by:

$$tf_{i,j} = \frac{n_{i,j}}{\Sigma_k n_{i,j}} \tag{1}$$

The inverse document frequency (IDF) computes the weights of the rare keywords across all URLs in our dataset. The rare keywords get high IDF scores and is given by:

$$idf(w) = log(\frac{N}{df_i}) \tag{2}$$

Therefore, TF-IDF is the combined score and is the product of TF and IDF. The TF-IDF score is given by:

$$w_{i,j} = tf_{i,j} \times log(\frac{N}{df_i}) \tag{3}$$

To make our text data informative for learning algorithms, we convert the keyword data into a vector of numbers using term-frequency inverse-document frequency (TF-IDF).

C. Keywords Clustering

We use k-means clustering to group similar keywords in our URL dataset. The k-means clustering algorithm is an unsupervised learning technique that allows us to identify similar groups or clusters of data points in our keywords dataset. We use the elbow method [12] to select the value of k. The elbow method provides an optimal value of k (the number of clusters) for our dataset. Elbow method is used to quantify the quality of clustering using the within-cluster SSE (sum of the squared differences between each observation and its group's mean.) or distortion. We use distortion to quantify the quality of the clusters.

D. Probabilistic Scoring Model

In this section, we design a probabilistic model by measuring the frequency of malicious keywords present in our test URL dataset. We do not have any prior knowledge about the test dataset regarding the presence of malicious keywords. We use our training model and the associated keywords' corpus to form the malicious keywords clusters. We then collect the top 100 terms from each cluster based on the measured TF-IDF score. To find the probabilistic score of the malicious keywords in the new URL dataset, we check the frequency of top terms in each URL in the test set. Based on the malicious term frequency, we compute a cumulative score for each URL in the test set as:

$$\frac{1}{n}[\frac{q_1}{k_1} + \frac{q_2}{k_2} + \cdots + \frac{q_n}{k_n}] * 100 \tag{4}$$

where n is the number of clusters, with $k_1, k_2, \cdots, k_n$ malicious keywords in each cluster, and $q_1, q_2, \cdots, q_n$ are the frequency of malicious keywords present in each URL. Based on this score we determine the maliciousness of each of the 500 URLs.

V. Preliminary Results and Discussion

Figure 2 shows the elbow method for evaluating the number of clusters, k. We compute the distortion by varying the number of clusters, k, from 1 to 300 over 1000 iterations for each k. From the figure, we choose $k = 35$ as we observe limited gains beyond this point.

We also evaluate the performance of clusters for our URL dataset. It is important to evaluate the quality of clusters for an unsupervised learning task as we do not have prior ground truth information. We use the silhouette analysis model to evaluate the performance of the clusters. Silhouette analysis [13] is an intrinsic metric used to evaluate the quality of clusters. We use it to quantify the quality of the clusters. We compute the silhouette coefficient for our URL keywords dataset to evaluate the performance of clusters. The silhouette coefficient is defined for each sample in our dataset and is the combination of two scores a and b. The term a is the mean distance between a given sample and other data points belonging to the same cluster, and b is the average distance between a given sample and all data points in the nearest neighboring cluster. Thus, for a single sample the silhouette coefficient is given by:

$$s = \frac{b - a}{max(a, b)} \tag{5}$$

Silhouette analysis is also a graphical tool for understanding the performance of k-means clustering. A plot of silhouette cofficient vs. the number of clusters is shown in Figure 3. From the figure, we can observe the size of the clusters and identify their quality. Also from the plot, we can measure how tightly the data points are grouped in the clusters. The silhouette coefficient varies between -1 and +1; -1 indicates incorrect clustering and +1 indicates highly dense clusters. The silhouette coefficient score around zero indicates overlapping clusters. The silhouette coefficient for our dataset is 0.383 and is an indicator of dense clustering.

Lastly, we also evaluate the test set for the presence of malicious keywords using the probabilistic model described in Section IV-D. A histogram of the score distribution for 500 URLs is shown in Figure 4. From our analysis, we observe that the frequency of malicious keywords varies from $0 - 14\%$ in the test set of 500 URLs. Based on our analysis, we determine that 34 URLs do not contain any malicious keywords. We

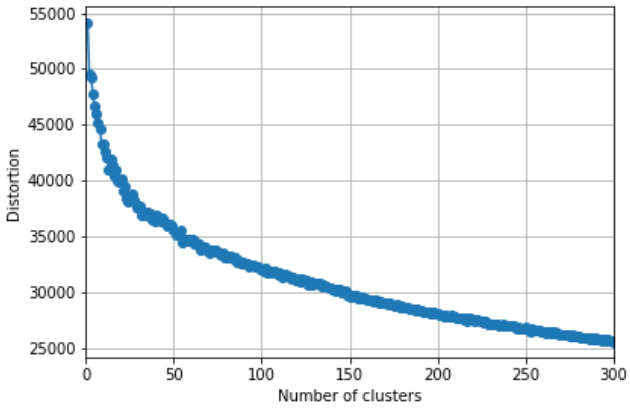

Fig. 2. Selecting k using the elbow method.

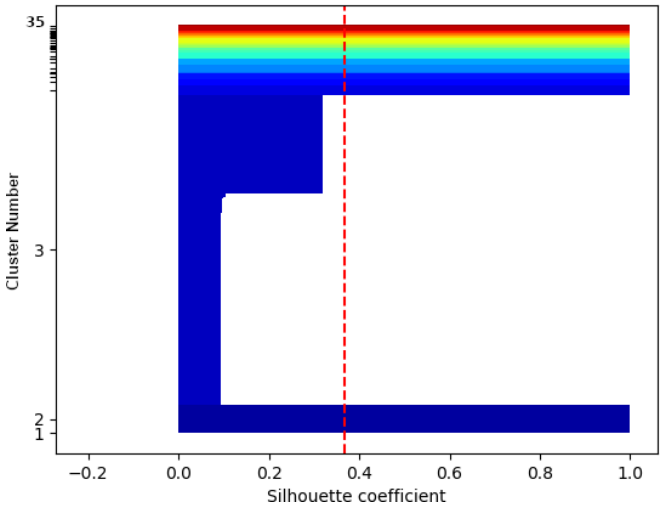

Fig. 3. Silhouette analysis for cluster quality evaluation.

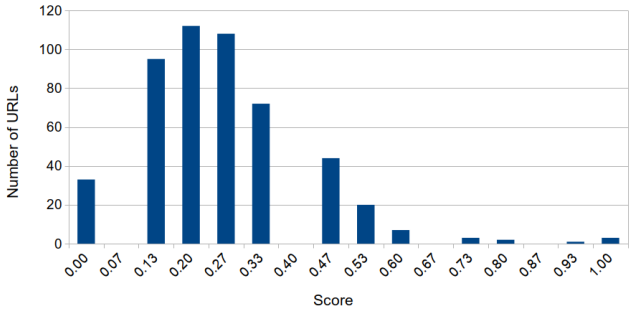

Fig. 4. Test set score distribution histogram.

note that URLs with scores above 0.08 contain at least one malicious keyword. Thus, our model can be effectively used to identify malicious URLs.

VI. Conclusion

In this paper, we proposed a URL keyword clustering approach to analyzing malicious URLs. We perform extensive data processing for sanitizing, tokenizing, and vectorizing the URL dataset. We demonstrated an approach based on k-means clustering for grouping malicious URL data. Our solution can be employed to detect malicious URLs from open source threat intelligence feeds. Our proposed unsupervised learning technique is effective in discovering hidden structures in the dataset. We use the elbow method to aid in model selection and choose $k = 35$ based on our evaluations. We also show that the clusters obtained using our approach are of good quality and have a silhouette coefficient of 0.383 for a dataset containing over 11,000 malicious URLs. We develop a probabilistic model to calculate the percentage of malicious keywords present in any given URL by comparing it with the top terms in the obtained clusters. We also developed a probabilistic model to detect new malicious URLs based on existing keyword patterns. After analyzing over 72,000 malicious keywords, our model successfully identifies over 80% of the URLs in a test dataset as malicious. Our future work will focus on developing hierarchical clustering algorithm for our dataset.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Number CNS-1817105. This work was completed using the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative.

References

[1] "Threat intelligence system," https://www.recordedfuture.com/threat-intelligence-initiatives/.
[2] "scikit-learn feature extraction from text," https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.
[3] "K-means algorithm," https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.
[4] E. Nunes, A. Diab, A. Gunn, E. Marin, V. Mishra, V. Paliath, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian, "Darknet and deepnet mining for proactive cybersecurity threat intelligence," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, Sep. 2016, pp. 7–12.
[5] S. Qamar, Z. Anwar, M. A. Rahman, E. Al-Shaer, and B.-T. Chu, "Data-driven analytics for cyber-threat intelligence and information sharing," *Computers & Security*, vol. 67, pp. 35–58, 2017.
[6] S. Brown, J. Gommers, and O. Serrano, "From Cyber Security Information Sharing to Threat Management," in *Proceedings of the 2Nd ACM Workshop on Information Sharing and Collaborative Security*, ser. WISCS '15. New York, NY, USA: ACM, 2015, pp. 43–49. [Online]. Available: http://doi.acm.org/10.1145/2808128.2808133
[7] "CRITs - Collaborative Research Into Threats," https://crits.github.io/.
[8] A. Modi, Z. Sun, A. Panwar, T. Khairnar, Z. Zhao, A. Doup, G. Ahn, and P. Black, "Towards automated threat intelligence fusion," in *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, Nov 2016, pp. 408–416.
[9] J. Navarro, V. Legrand, S. Lagraa, J. François, A. Lahmadi, G. De Santis, O. Festor, N. Lammari, F. Hamdi, A. Deruyver, Q. Goux, M. Allard, and P. Parrend, "Huma: A multi-layer framework for threat analysis in a heterogeneous log environment," in *Foundations and Practice of Security*, A. Imine, J. M. Fernandez, J.-Y. Marion, L. Logrippo, and J. Garcia-Alfaro, Eds. Cham: Springer International Publishing, 2018, pp. 144–159.
[10] S. Lee and T. Shon, "Open source intelligence base cyber threat inspection framework for critical infrastructures," in *2016 Future Technologies Conference (FTC)*, Dec 2016, pp. 1030–1033.
[11] Z. Guan, L. Bian, T. Shang, and J. Liu, "When machine learning meets security issues: A survey," *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, pp. 158–165, 2018.
[12] D. J. Ketchen and C. L. Shook, "The application of cluster analysis in strategic management research: an analysis and critique," *Strategic management journal*, vol. 17, no. 6, pp. 441–458, 1996.
[13] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2014.