Poster: On the Application of NLP to Discover Relationships Between Malicious Network Entities

Giuseppe Siracusano
NEC Laboratories Europe
giuseppe.siracusano@neclab.eu

Martino Trevisan
Politecnico di Torino
martino.trevisan@polito.it

Roberto Gonzalez
NEC Laboratories Europe
roberto.gonzalez@neclab.eu

Roberto Bifulco
NEC Laboratories Europe
roberto.bifulco@neclab.eu

ABSTRACT

The increase in network traffic volumes challenges the scalability of security analysis tools. In this paper, we present NetLearn, a solution to identify potentially malicious network entities from large amounts of network traffic data. NetLearn applies recently developed natural language processing algorithms to discover security-relevant relationships between the observed network entities, e.g., domain names and IP addresses, without requiring external sources of information for its analysis.

ACM Reference Format:
Giuseppe Siracusano, Martino Trevisan, Roberto Gonzalez, and Roberto Bifulco. 2019. Poster: On the Application of NLP to Discover Relationships Between Malicious Network Entities. In *2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19), November 11–15, 2019, London, United Kingdom.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3319535.3363276

1 INTRODUCTION

Network traffic volumes have been growing at exponential rates, and new network technologies, such as 5G, are expected to further speed up such growth. Performing complex security analysis on the entire volume of data is usually not possible or anyway expensive. This paper presents a scalable solution to identify potentially malicious network entities, i.e., domain names and IP addresses, in large volumes of network traffic. This allows network administrator to focus more expensive analysis techniques only on the subset of network traffic that contains the identified potentially malicious entities.

As shown in previous work [6], network entities contacted by a host may contain particularly useful information for security analysis. This helps a scalable analysis of the network traffic, since network entities can be determined by looking only at the first packets of a flow. Instead, other metrics, such as packet inter-arrival time, require per-packet processing, e.g., to compute values used to update per-flow counters. As a result, a system that monitors network destinations needs to scale with the number of flows in the network, instead of scaling with the number of packets, i.e., the traffic volume. For example, a single flow belonging to a video application may contribute thousands of network packets, but its volume would not affect the scalability of a security analysis that looks only at its network destination.

Nonetheless, a joint analysis of the observed network entities is usually required to derive security relevant information. For instance, a possible approach is to build a co-occurrence matrix of domain names[1], in order to discover network entities that are related to each other, such as those belonging to malware distribution chains or web sites [4]. A co-occurrence matrix is a square $n \times n$ matrix, with n being the number of observed network destinations. Each row of the matrix essentially tells the statistical relationship between the selected domain and all the other observed domains, ultimately allowing a security algorithm to reason about the relationship between malicious domains and unknown domains. Unfortunately, even medium-sized networks carry traffic to millions of different network entities, making the co-occurrence matrix quickly grow to sizes that are difficult to handle.

We design NetLearn to address this issue. NetLearn leverages recent advances in natural language processing to build an approximated version of the information contained in a co-occurrence matrix. More in detail, we train a skip-gram model that is able to build vector representations of network entities using an unsupervised learning process. The model is fed with sequences of domains (and server IPs) contacted by a network host during a certain time window, e.g., a day. NetLearn does not need to access the payload of packets, and, as such, works seamlessly with encrypted traffic. The model maps each entity to a point in an Euclidean space, and points close to each other represent semantically similar network entities, e.g., two malicious domains. This property allows us to cluster the obtained vectors. Then, using the information provided by a ground truth source, e.g., an existing blacklist, we label clusters containing malicious network entities, and identify the points belonging to the same clusters as likely malicious.

Our preliminary results show that NetLearn can identify new unknown malicious network entities, analyzing only a small subset of the observed network entities.

2 CONCEPT

Why does the co-occurrence relationship between network entities help discovering security-relevant information? Here we provide

[1] We informally use the term *domain* name throughout the paper meaning Fully Qualified Domain Name.

CCS '19, November 11–15, 2019, London, United Kingdom

ACM ISBN 978-1-4503-6747-9/19/11.
https://doi.org/10.1145/3319535.3363276

three examples that help the reader build an intuition about the information contained in such data.

Web sites profiling. Modern web sites include a number of external resources, hosted at domain names that may be different from the main web site's domain. For instance, [2] reports that the Alexa top 1M web sites have resources located on tens of different external domains, on average. The nature of different domains (e.g., news, e-commerce, advertisement, CDN, etc.), and the number of such domains contacted by a web site can provide insightful information about the nature of the web page itself.

Malware distribution chains detection. Previous work has measured the prevalence of an underground market of malware distribution chains [4]. Such chains are characterized by a landing web page that is legit, e.g., a compromised benign web page or a page hosting a third-party advertisement, which then initiates a series of automatic redirections towards a malware distribution web site that usually exploits some vulnerabilities to implement drive-by downloads. As such, the sequence of network destinations may once more help to identify such distribution chains.

Botnet detection. Recent work showed that botnets' hosts have a noisy network behavior [3]. For example, they assess connectivity, retrieve date information, and perform scanning to detect the command&control channel. Such activities generate a number of network flows towards different destinations. Such communication patterns may reveal important information to detect botnets. For example, dynamic analysis tools run malwares to analyze their network behavior and generate signatures that may help identify them.

From the discussion so far it seems clear that network flows' destinations can contain information helpful to detect potentially malicious network entities. Considering the examples previously discussed: web sites usually trigger several network flows while loading, in short sequence; distribution chains perform quick redirection to different web domains; and even botnet hosts may generate several network flows in a short time interval. Thus, if we further assume it is possible to distinguish the generated network flows on a per-host basis (e.g., using source IP addresses as host identifiers), it should be also possible to easily observe sequences of *related* network entities in the network traffic. Given the ability to obtain sequences of potentially related network entities, the challenge shifts to the implementation of a mean that could discover the meaning of such sequences.

2.1 NetLearn

How to identify the different meanings of different sequences? Our main intuition comes from the observation that learning the meaning of a sequence of related entities is a typical task performed by recent NLP algorithms. In particular, recent work has focused on the task of learning a mathematical representation of the meanings associated with single words. The process of learning is itself unsupervised, and performed by feeding the algorithm with a large corpus of sentences, i.e., sequences of words. Once the learning is done, the learned representations for the words, also called word embeddings, are vectors corresponding to points in an Euclidean space.

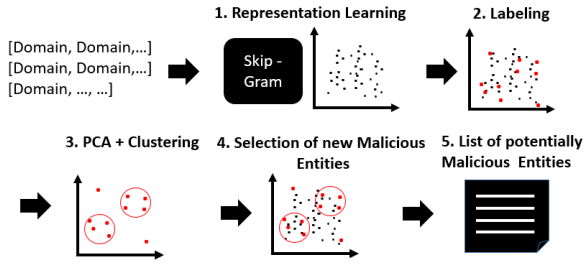

Figure 1: NetLearn operations.

Intuitively, given a definition of distance (Euclidean norm, cosine similarity, etc.), word vectors close to each other should have similar meanings. In addition, the learned embeddings capture relationships between word meanings on different levels, which go beyond the grouping of words with similar meanings. A typical example used in the area is related to the relationship between learned representations of the words "man", "woman", "king", "queen". Using NLP algorithms on a large enough *corpus* of sentences, it should be possible to learn representations of the above words, such that the mathematical relationship between the vectors representing "man" and "king" is the same that holds between the vectors representing "woman" and "queen".

In NetLearn we use the same algorithms that are used to learn the meaning of words, but to instead capture the network entities' "meanings". That is, by replacing words with network domain names and IP addresses, and sentences with network sessions, we are able to apply NLP techniques to represent domains as vectors, therefore approximate the information that would be contained in a co-occurrence matrix. The obtained vector representations can then be clustered, and network entities clustered with known malicious network entities can be marked as suspicious for further analysis.

2.2 Operations overview

Figure 1 briefly shows NetLearn operations, whose description follows. NetLearn assumes that sequences of domain names or server IP addresses are provided as input to the system. These are usually easily extracted from network traffic, log files, etc. We assume each network host in the trace to be associated with its sequence of contacted domain names and IP addresses, ordered according to the time of appearance in the network.

Step 1. The corpus of sequences is provided offline for training. We expect this training to be performed periodically, as new traffic traces are collected, in order to refine the learned vector representation and to update them as new data becomes available.

Step 2. The learned vector representations are labeled as BAD using a source of ground truth. We use existing blacklists as ground truth. Most of the time, only a very small subset of vectors will be labeled as BAD, however, clusters of BAD vectors may indicate that the algorithm was able to learn some common properties associated with those entities. As such, other unlabeled vectors close to such clusters may share common properties with BAD vectors, for which we may suspect them to be unknown malicious entities.

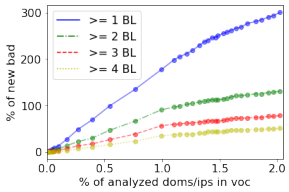

Figure 2: Percentage of newly discovered malicious entities, using CBL as labeling ground truth.

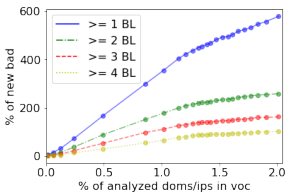

Figure 3: Percentage of newly discovered malicious entities, using GSB as labeling ground truth.

Step 3. We run a clustering algorithm on the subset of BAD vectors. This allows us to quickly perform the clustering, since we try to cluster only the (few) BAD vectors, and to discover which ones among them share common properties.

Step 4. We examine all BAD vectors belonging to a cluster, and select the nearby unlabeled vectors to add them to a list of suspicious entities. The selection process may happen in different ways, e.g., using different concepts of distance, as detailed later.

3 EVALUATION

We performed a preliminary evaluation of `NetLearn` using a dataset from an operational network of an European Internet Service Provider. The dataset includes anonymized traffic from about 2,000 broadband users, and one week of data captured in 2018. Tstat [5] was used to capture network traffic data. Flows of the same client contribute to build a sequence, which is terminated after 2 minutes of inactivity.

We use two different blacklists as ground truth: Google Safe Browsing (GSB), and a widely used commercial blacklist (CBL). To confirm the nature of an entity identified by `NetLearn` as potentially malicious we use VirusTotal [1]. Since VirusTotal reports the number of blacklists/products in which a network entity was signaled as malicious, we use such number to increase the confidence about the maliciousness of a given entity. That is, in different tests we consider malicious an entity if 1, 2, 3 or 4 VirusTotal's sources report it as malicious.

Preliminary results. `NetLearn` marks as potentially malicious a network entity if its vector representation has a similarity value with a known BAD domain higher than a given threshold. I.e., if the threshold is 0, the entire dataset would be marked as malicious. During the tests we adjust this threshold to trade-off the number of newly discovered malicious network entities with the number of entities that need to be checked with, e.g., VirusTotal. Figures 2 and 3 show the percentage of new malicious network entities discovered, normalized to the size of the used ground truth, when varying the threshold value to analyze from 0 to 2% of the dataset. For instance, to provide an over 100% increase of the blacklist, `NetLearn` needs to analyze less than 0.6% and 0.25% of all entities, when using CBL and GBS as ground truth, respectively.

4 DISCUSSION

Our preliminary results point to a number of interesting directions for improvements and further research. First, we believe that `NetLearn` works better for the identification only of a subset of the several categories of malicious network domains and IP addresses. We plan to perform a number of targeted tests to identify for which categories `NetLearn` may be best suited, for instance adopting blacklists containing only a type of malicious entities as ground truth. This would also allow us to compare `NetLearn` with alternative security analysis tools, in order to perform a cost/benefit evaluation with other approaches. Second, we plan to study multiple datasets, to verify that our findings hold true also in heterogeneous environments. Finally, we plan to explore newer NLP algorithms which extend the concept of context of an entity, taking into account also the ordering of the entities in a sequence and capture more complex relationships between entities.

ACKNOWLEDGMENTS

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 761592 ("5G ESSENCE"). This paper reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] 2019. Virus Total. https://www.virustotal.com.
[2] Deepak Kumar, Zane Ma, Zakir Durumeric, Ariana Mirian, Joshua Mason, J Alex Halderman, and Michael Bailey. 2017. Security challenges in an increasingly tangled web. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 677–684.
[3] Chaz Lever, Platon Kotzias, Davide Balzarotti, Juan Caballero, and Manos Antonakakis. 2017. A lustrum of malware network communication: Evolution and insights. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 788–804.
[4] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. Shady paths: Leveraging surfing crowds to detect malicious web pages. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 133–144.
[5] Martino Trevisan, Alessandro Finamore, Marco Mellia, Maurizio Munafo, and Dario Rossi. 2017. Traffic analysis with off-the-shelf hardware: Challenges and lessons learned. *IEEE Communications Magazine* 55, 3 (2017), 163–169.
[6] Yury Zhauniarovich, Issa Khalil, Ting Yu, and Marc Dacier. 2018. A survey on malicious domains detection through DNS data analysis. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 67.