port2dist: Semantic Port Distances for Network Analytics

Laurent Evrard
University of Namur, Namur, Belgium
laurent.evrard@unamur.be

Jérôme François
Inria, Nancy, France
jerome.francois@inria.fr

Jean-Noël Colin
University of Namur, Namur, Belgium
jean-noel.colin@unamur.be

Frédéric Beck
Inria, Nancy, France
frederic.beck@inria.fr

***Abstract*—Traffic analysis is a predominant task to support multiple types of management operations. When shifting from manually built signatures to machine learning techniques, a problem resides in the model to represent traffic features. The most notable examples are the TCP and UDP ports, near port numbers in the numerical space is not representative of a close semantic from an operational point of view. We have thus developed a technique to learn meaningful metrics between ports from scanning strategies followed by attackers. In this demonstration, we propose the *port2dist* tool, allowing to get, seek and retrieve semantic dissimilarities between port numbers.**

I. Introduction

Most of IP-based communications occur with TCP or UDP. Those protocols rely on port numbers to de-multiplex flows between services. In the era of artificial intelligence and machine learning, comparing or evaluating distances between network flows or packets may rely on the nature of the services they convey, and so on the associated port numbers.

While using port numbers assigned by the Internet Assigned Numbers Authority (IANA) is not mandatory, their use eases the access to the service. Although port numbers are numerical features, they cannot be mapped directly to a meaningful metric space. Assuming the examples with three TCP ports: 25 (SMTP), 80 (HTTP) and 443 (HTTPS), it sounds more legitimated to consider HTTP and HTTPS closer than SMTP and HTTP while numeric values does not reflect it.

We do not claim that there is only one good reasoning regarding the similarity between port numbers. However, analyzing how the attackers target the different ports allows to infer a strategy that is associated to a certain meaning.

In [1], by observing attackers at a large scale, we propose a similarity measure between TCP port numbers that is able to catch two types of semantics:

- Service-semantic similarity for ports used by similar services like 80 and 443 in the previous example
- Context-semantic similarity for ports usually used by the same host or in a close vicinity such as a web (HTTP) and database (3306 - MySQL) server

Full details about the theoretical definition of the metric is given in [1]. This current paper focuses on the architecture of our prototype, *port2dist*, which is in charge of collecting and modeling darknet data, extracting similarities between port numbers and providing those results through a public API. Any user interested in analyzing data containing TCP port numbers can thus use it and take benefit of the knowledge contained in our darknet capture for his or her own analysis. The main advantage of this approach is to provide a new tool for any user without disclosing original data (darknet network traffic) whose the access is under strict restrictions.

978-3-903176-15-7 © 2019 IFIP

II. Similarity between TCP port numbers

Unlike IP addresses that can be represented in a hierarchical manner with subnetworks [2] or using embedding techniques [3], a few has beed proposed to compare port numbers in a meaningful manner. As a result, machine learning algorithms using port numbers raise issues [4]. In [5], the authors aim to identify a predominant port number to characterize a group of flows. In [6], a large-grained approach is leveraged by considering the three ranges defined by the IANA: registered, well-known or dynamic.

We propose a network dissimilarity metric (called distance for the sake of simplicity) built from darknet (or network telescope) observations. In a darknet, an entire unused subnet is reserved to silently collect all received unsolicited IP traffic.

With our /20 darknet, up to 48 millions of TCP SYN packets are observed every day. Many of them are related to either full vertical or full horizontal scans but some of them reveal a particular attacker strategy probing carefully-selected port numbers. Our objective is so to extract the distances between port numbers from these finely tailored scans. After extracting all the sequences of the representative scanned ports, they are aggregated into a unique weighted graph representing the transitions between the targeted ports. This graph allows to encode probing activities in order to convey trends and knowledge of the attackers when they scan IP addresses to locate potential targets.

For example, the edges from/to 80 from/to 443 present a heavy weight because they are frequently scanned within the same sequence. After some post-processing (rescaling, inversion), shortest distances between port numbers represent their distances from a semantic point of view. As explained in [1], smart attackers probe ports, which are in their interest (same type of services) or co-located together to make their scans more efficient and so stealthier. This intrinsically reveals both the service- and context-semantic similarity respectively.

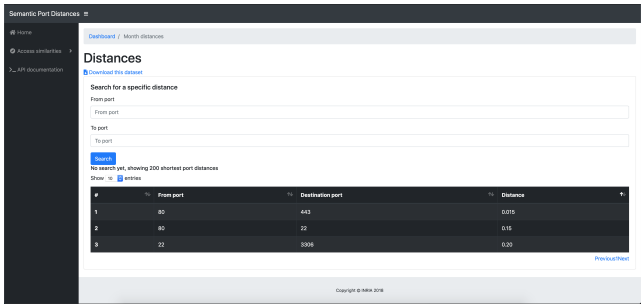

Fig. 1: Screenshot of the web interface

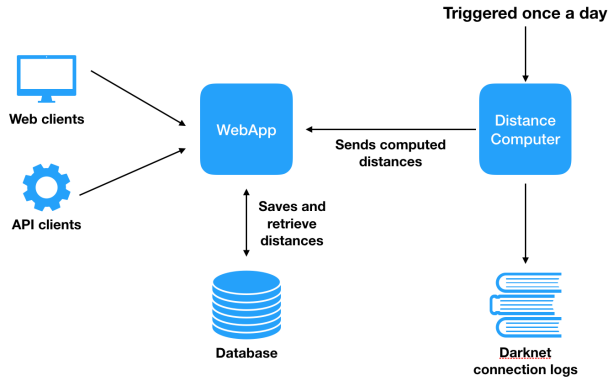

Fig. 2: Architecture of the application

III. Application and scenario description

A. Features and interfaces

We developed a web platform, *port2dist*, allowing anyone to fetch semantic distances between network ports. Each day, semantic distances are computed for the last day, week and month. The same applies every month for the last month and year. It creates a moving time window allowing users to select the most appropriate time slice for her analysis purpose. The dataset can be downloaded as a whole or specific port numbers can be searched.

Two interfaces are exposed to provide the aforementioned functionalities: a web interface illustrated in figure 1 and a REST API. The documentation of the REST API is provided in the web interface.

B. Architecture

The architecture of *port2dist* is composed of two components, presented in figure 2. First, the *Distance Computer* is triggered once a day to compute the four new distances (day, week, month, year) from darknet data. It uses the pipeline presented in section II. Second, once the distance generation phase is finished, they are sent to the *Web App* component. The latter stores a complete version (all computed distances) which can be then downloaded as a whole by users or requested for chosen ports through the REST API.

C. Demonstration scenario

The demonstration will consist in two phases. First, the application and its web interface will be introduced and we will show how to use it and the calculated distance for representative examples. For example, web-based ports are singular examples as the are very close to each others.

Second, the use case of predictive TCP scan blocking tool will be demonstrated. Once a first scan is detected, it consists in automatically blocking the next ports that will be targeted by the scans. These ports are inferred from the knowledge given by the previously defined distance.

Although full results are detailed in [1] with regular metrics such as true positives or false positives, the goal of the demonstration is to show how decisions are made by our analysis engine. Hence, a network trace containing TCP scans will be replayed at a small pace to show how the blocking policy is built in real-time.

IV. Conclusion

This demonstration highlights how the probing of TCP ports observed at large scale actually gives the security analyst an unique opportunity to counteract against attacks. Our inter-port distances are data-driven defined and built for data analysis purposes. The distances do not need to be computed and applied with the same data source as our port blocking application demonstrates. It has been proved to be highly effective in [1] and we thus decide to make this consolidated knowledge public (http://port2dist.lhs.inria.fr/). Therefore, we have extended our prototype with a web and REST interface as well as with the possibility to extract the distances for a particular time window.

Acknowledgments This work has been partially supported by the project SecureIoT, funded from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 779899 and by the NATO Science for Peace and Security Programme under grant G5319 Threat Predict: From Global Social and Technical Big Data to Cyber Threat Forecast. It is also supported by the High Security Lab (https://lhs.loria.fr).

References

[1] L. Evrard, J. François, and J.-N. Colin, "Attacker Behavior-Based Metric for Security Monitoring Applied to Darknet Analysis," in *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management*, Washington, DC, USA, 2019. [Online]. Available: http://port2dist.lhs.inria.fr/static/im2019.pdf

[2] L. Dolberg, J. François, and T. Engel, "Efficient Multidimensional Aggregation for Large Scale Monitoring," in *Large Installation System Administration Conference (LISA)*. San Diego, USA: USENIX, 2012. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00784953

[3] M. Li, C. Lumezanu, B. Zong, and H. Chen, "Deep learning ip network representations," in *SIGCOMM Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, ser. Big-DAMA. ACM, 2018.

[4] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[5] L. Grimaudo, M. Mellia, E. Baralis, and R. Keralapura, "Select: Self-learning classifier for internet traffic," *Transactions on Network and Service Management*, vol. 11, no. 2, pp. 144–157, June 2014.

[6] S. E. Coull, F. Monrose, and M. Bailey, "On measuring the similarity of network hosts: Pitfalls, new metrics, and empirical analyses," in *Network and Distributed System Security Symposium*, 01 2011.