A Scalable Network Event Detection Framework for Darknet Traffic

Max Gao, Ricky K. P. Mok, kc claffy
CAIDA/UC San Diego

ACM Reference Format:
Max Gao, Ricky K. P. Mok, kc claffy. 2022. A Scalable Network Event Detection Framework for Darknet Traffic. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3517745.3563015

Extended Abstract

Unsolicited network traffic captured by network telescopes, namely *darknet traffic*, provides important data for studying malicious Internet activities, such as network scanning [9], the spread of malware [4], and DDoS attacks [6]. Inferring such activity in traffic often requires first obtaining fingerprints of the activity and searching historical traffic traces (e.g, pcaps) for that pattern. Traffic volume at the largest darknets can exceed 100GB/hour, rendering it challenging to process at the packet level. Aggregated flow-based metadata [2] can reduce computation, storage and I/O overhead at the expense of finer-grained information about the traffic. Customized data structures (e.g., [7]) and streaming algorithms (e.g., [5]) offer an alternative approach to extracting information from raw packets, but they are typically traffic tailored for estimating specific metrics and thus limited in their ability to detect a wide range of events.

We propose a machine learning (ML)-based framework to detect events by characterizing traffic dynamics across many time series generated from raw traffic processed by the Corsaro software package [1]. Our method extracts signals of attacks in time-series statistics that can reveal promising time periods in which to further investigate an attack using raw packet traces.

Methodology

Our framework leverages the time-series traffic metrics (feature dimensions) generated by the Corsaro software suite (Fig. 1). Corsaro tabulates five traffic metrics every minute based on six properties of incoming packets collected by the telescope (Table 1) and stores them into InfluxDB, a time series database. The properties cover network protocol information from packet headers, metadata inferred using prefix-to-AS datasets, IP geolocation databases, and spoofing classification [3]. Each combination of the properties results in a distinct time series, yielding over 200K time series per week (accessible at https://explore.stardust.caida.org).

Our ML-based analysis component queries the time series data from InfluxDB and captures three major types of events:

I. *Repeated events* in the same time series could reveal re-use of similar attack techniques/tools over time.

Table 1: Traffic metrics and properties whose combinations produce observations (per minute) yielding over 200K time series.

Properties	Metrics (per minute)
Origin ASN	# of packets (PPM)
Geolocation	# of bytes (BPM)
Protocol number	# of unique source IPs
TCP/UDP Destination port	# of unique source ASN
ICMP type & code	# of unique destination IPs
Spoofing inference	

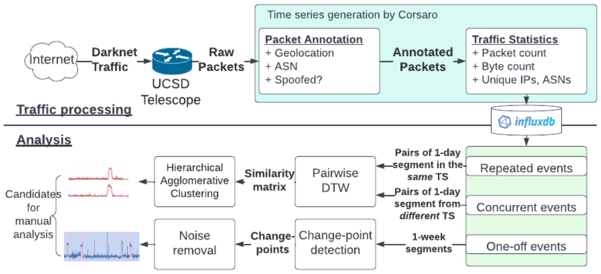

Figure 1: Data Analysis Architecture outputs a list of events that merit deeper analysis.

II. *Concurrent events* across different time series could reveal fingerprints and scale of network attacks.

III. *One-off events* refers to significant surges in traffic with the same proprieties.

Type I & II rely on the same analysis method different time series segments as input to the ML model. We often do not know the nature of the traffic activity before analysis, so we adopt an unsupervised approach to learning similarities in traffic patterns.

We first partition time series of T observations belonging to a set of traffic properties (Table 1) into N segments of length b. We denote a z-normalized segment n of the partitioned set of time series for a given metric m from a set of metrics M as $\{X_m^n\}_{m \in M, n \in N}$. We then apply Dynamic Time-Warping (DTW) [11] to compute a similarity measure between any two combinations of n, storing them in a symmetric distance matrix, $D \in \mathbb{R}^{n \times n}$, for type I event analysis. DTW generalizes to multiple dimensions and could compute more semantically precise results by combining time series across multiple traffic metrics. On the other hand, combining time series across multiple traffic properties yields type II events.

The second step is to cluster segments based on their similarities within the distance matrix. DTW restricts our choice of clustering algorithm to ones that do not assume inputs are located in Euclidean space (DTW produces measures that violate the triangle inequality). We chose Single-Linkage Hierarchical Agglomerative Clustering (HAC) [10] for segment clustering. HAC iteratively forms clusters: in each iteration, a cluster is merged with another if any of their members share a minimum distance. Prior to any merges, HAC treats each input as a singleton cluster.

IMC '22, October 25–27, 2022, Nice, France
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9259-4/22/10.
https://doi.org/10.1145/3517745.3563015

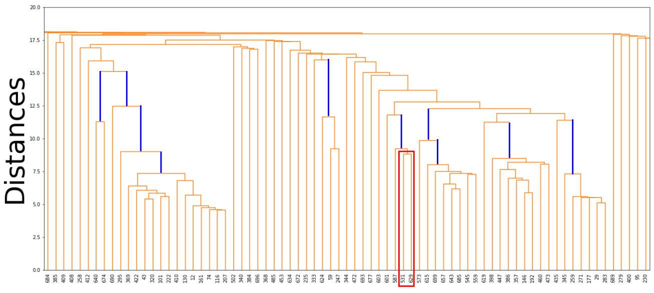

Figure 2: Partial dendrogram of clustered U.S. PPM segments. Some clusters have large cophenetic distances (blue lines).

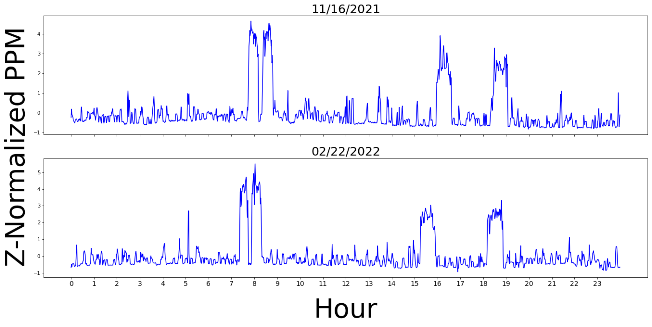

Figure 3: Two segments belonging to the cluster formed at the first step of HAC (marked with red in Fig. 2). Both time series possessed similar z-normalized PPM values throughout the day.

For type III, we employ change point detection algorithms (CPDs) to detect level shifts from the baseline contained in a segment. We evaluated out-of-the-box performance of cost-function based CPDs [8] using 1-week time series segments and found that the results contained many false positives. We apply smoothing and quartile filtering to reduce noise, noting that their accuracy may generalize better when applied to segments of select clusters.

Preliminary results

We present preliminary results of detecting type I events from 2 years (June 3, 2020 - June 3, 2022) of UCSD Telescope data. We selected 85 time series for analysis: all 5 metrics for each time series with traffic source geolocated to 17 countries. We chose a segment size of 1 day (b=1440 min) to capture diurnal patterns of unique source IPs and ASNs. A smaller b could enable finer-grained comparison, but reduce our confidence in the z-normalized values (due to fewer observations). The value of b does not affect the number of computations but it may affect execution time depending on the implementation.

Fig. 2 shows the HAC clustering results of the distance matrix computed from PPM metrics daily segments for source IPs geolocated to the U.S. Fig. 3 shows two initially merged singleton clusters belonging to the dendrogram. Although the two days were 3 months apart, their patterns were similar. The second subsequent merge's cophenetic distance is large compared to the first and to merges across the tree. Large cophenetic distances imply that the two adjoined clusters should remain separate. Assessing the cophenetic distance distributions of a dendrogram offers insight into possible cutoff values for defining a 'true' cluster. High variance in the distribution relates to the sparsity in a dendrogram's tree, which reflects the degree of heterogeneity across time series segments. Distinctions in sparsity are visually observable across the dendrograms pertaining to different countries' metrics, implying that each dendrogram requires a different cophenetic cutoff value.

Once clusters are defined, we may choose segments to serve as a representation of the baseline and quantify their proportion to aberrant traffic segments. Comparing new segments with existing clusters functions as a method for detecting anomalies.

We ran our analysis on Expanse [12], an advanced HPC system at UC San Diego's Supercomputing Center. Using jobs configured with 64 CPU cores, the distance matrix computation time was less than 6 hours.

Conclusion and Future Work

We examined the feasibility of ML approaches for time series analysis to detect network events in darknet traffic. Our results showed preliminary success in identifying repeated events in two years of UCSD Network Telescope data. Next, we will evaluate the accuracy of the framework in terms of identifying different types of network activities. We will study traffic data during the events and refine automated approaches to their detection. We will create datasets for predictive model training purposes, amenable to cross-evaluation with other large-scale datasets.

Acknowledgements

This work is based on research sponsored by the National Science Foundation grants CNS-2120399, CNS-2131987, CNS-1730661, and the Amateur Radio Digital Communications (ARDC) Foundation. The views are those of the authors and do not necessarily represent endorsements, either expressed or implied, of the sponsors.

References

[1] 2012. Corsaro. https://catalog.caida.org/media/2012_dust_corsaro.
[2] Shane Alcock. 2021. Flowtuples IV: Reality Strikes Back. https://catalog.caida.org/media/2021_flowtuples_iv_dust.
[3] Alberto Dainotti, Karyn Benson, Alistair King, and others. 2014. Estimating Internet Address Space Usage through Passive Measurements. *SIGCOMM Comput. Commun. Rev.* 44, 1 (dec 2014), 42–49.
[4] Alberto Dainotti, Alistair King, et al. 2012. Analysis of a "/0" Stealth Scan from a Botnet. In *Proc. ACM IMC.*
[5] Stefan Heule, Marc Nunkesser, and Alex Hall. 2013. HyperLogLog in Practice: Algorithmic Engineering of a State of The Art Cardinality Estimation Algorithm. In *Proc. ACM EDBT.*
[6] Mattijs Jonker, Alistair King, Johannes Krupp, and others. 2017. Millions of Targets under Attack: A Macroscopic Characterization of the DoS Ecosystem. In *Proc. ACM IMC.*
[7] Jeremy Kepner, Michael Jones, Daniel Andersen, and others. 2021. Spatial Temporal Analysis of 40,000,000,000,000 Internet Darkspace Packets. In *Proc. IEEE HPEC.*
[8] R. Killick, P. Fearnhead, and I. A. Eckley. 2012. Optimal Detection of Changepoints With a Linear Computational Cost. *J. Amer. Statist. Assoc.* 107, 500 (oct 2012), 1590–1598.
[9] D Moore, C Shannon, G Voelker, and S Savage. 2004. *Network Telescopes: Technical Report.* Technical Report. CAIDA.
[10] Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. https://doi.org/10.48550/ARXIV.1109.2378
[11] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, and others. 2012. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proc. ACM SIGKDD.*
[12] Shawn Strande, Haisong Cai, Mahidhar Tatineni, and others. 2021. Expanse: Computing without Boundaries. In *Practice and Experience in Advanced Research Computing.* ACM.