Attacker Behavior-Based Metric for Security Monitoring Applied to Darknet Analysis

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

***Abstract*—Network traffic monitoring is primordial for network operations and management including Quality-of-Service or security. One major difficulty when dealing with network traffic data (packets, flows, etc) is the poor semantic of individual attributes (number of bytes, packets, IP addresses, protocol, TCP/UDP port numbers, etc). Many of them can be represented as numerical values but cannot be mapped to a meaningful metric space. Most notably are application port numbers. They are numerical but comparing them as integers is meaningless. In this paper, we propose a fine grained attacker behavior-based similarity metric allowing traffic analysis to take into account semantic relations between port numbers. The behavior of attackers is derived from passive observation of a darknet or telescope, aggregated in a graph model, from which a dissimilarity function is defined. We demonstrate the veracity of this function with real world network data in order to pro-actively block 99% of TCP scans.**

I. Introduction

TCP and UDP are major transport protocols in Internet. Port numbers allow the end-hosts to de-multiplex flows and forward them to the right sockets and so services. Being encoded in 16 bits, there are 65,536 possible ports for each TCP and UDP. There are different segments: system or well-known ports (0-1023), reserved ports for specific applications or vendors (1024-49151) and dynamic ports (49152-65535). Although the dynamic ports are mainly used as ephemeral ports, such as source ports when establishing a connection, other ports are associated to a special use, *i.e.* service. Their numbering is managed by the Internet Assigned Numbers Authority (IANA). Even if users are free to use any port numbers, using assigned port numbers eases access to the service.

They are a valuable source of information for managing and operating a network as for instance to perform traffic engineering for QoS purposes or to detect anomalies [1]. In many cases, packets or flows need to be compared for supporting machine learning or data-mining algorithms. For example, Netflow records can be analyzed to detect anomalies [2] but all flow attributes cannot be represented in a metric space to be easily compared. While using longest common prefixes can partially solve the problem with IP addresses [3], it remains valid for port numbers.

978-3-903176-15-7

In this paper, we propose an automated fine-grained approach to catch simultaneously two types of similarities between port numbers:

- Service-semantic similarity: this represents port numbers supporting services of the same type. For instance, TCP ports 80 and 443 are semantically close to each other (Web). However, TCP ports 443 and 22 are also close semantically because they provide a secure connection.
- Context-semantic similarity: this abstracts the relations between ports which are often present together (on the same machine or in a close vicinity, e.g. same sub-network). As an example, an medium-scale enterprise network often provides a web and email server.

It is worth to mention that two ports can be similar on both perspective, *e.g.* 443 and 80, both for web services and usually co-located on the same server. In a preamble of an attack, port scanning is often performed to find open ports. In order to remain undetected, attackers may prefer to target particular ports rather than using massive scans. Actually, the selection of these ports follows a logic that can be guided by a motivation equivalent to the service- or context-semantic. As a result, observing the port scan strategies performed by the attackers is helpful to derive the semantic between port numbers. Three contributions are presented in this paper. Firstly, major trends on port scanning are highlighted from a 40 weeks long darknet dataset. A darknet refers here to an unused IP subnetwork passively collecting incoming unsolicited traffic. It results in the clear observations of relationships among targeted ports. Secondly, this motivates and guides the definition of a metric that is defined in order to catch simultaneously both types of similarities (service- and context- semantic), based on a previous work [4]. Finally, our third contribution leverages our metric to proactively block scanned ports.

The remainder of the paper is structured as follows: Section II presents related works. Section III introduces our attacker-based semantic port similarity. Section IV details observations from our darknet. Our proposed metric is then evaluated in Section V and applied to real world internet traffic use case in Section VI. Section VII finally gives the conclusion with possible future work.

II. Related work

TCP/UDP ports used by applications is helpful for traffic monitoring purposes. However, some researchers like in [5]

question the use of the latter in machine learning methods because of the versatility of this information. In 2005, the authors already show that 70% of the network traffic can be properly classified based on official port numbering [6]. Nowadays, traffic classification is even facing new challenges with encrypted traffic [7]. However, in many cases, port numbers bring valuable information about the type of services in use or targeted by the attackers. We have shown that there are particular relationships between the sequences of scanned ports [4]. Actually, the weakness of using port numbers in data analysis is the lack of a proper metric to apprehend the similarity or dissimilarity between them since they are not embedded in a metric space. Many traffic analysis techniques rely thus on other features [8] or simply consider if port numbers are equal or not [9], [10]. In [11], the aim is to group TCP flows in order to identify a dominant port per group if it exists. In [12], port numbers are compared accordingly to the ranges they belong to (registered, well-known or dynamic). We propose to go further by deriving a single inter-TCP ports similarity metric.

Our similarity relies on knowledge indirectly embedded in the attacker activities, especially the TCP scans. Some surveys like [13], [14] show that large scanning campaigns become more frequent. Collecting scanning activities is thus a rich source information but necessitates a mining approach to extract synthetic knowledge. Darknets have been proved to be efficient to monitor large scale attacker activities such as scans or DDoS (Distributed Denial-of-Service) attacks [**?**], [15]. Many of existing works focus on analyzing and describing observations made through the darknet [16], [17]. This paper proposes to build a similarity function based on those observations to be then applied for real time security monitoring in another environment. In particular, we show the viability of our technique to pro-actively block future TCP scans. It is complementary to many existing techniques dealing with reactive detection of scans [18]–[20].

III. Attacker behavior-based inter-port measure

A. Rationale

The first stage of an attack usually consists in identifying the potential targets. Discovering accessible machines and services often relies on IP sweeping and/or scanning TCP and UDP ports [21]. Naive approach testing all ports numbers and all IP addresses of a targeted subnetwork is time-consuming and has a large footprint, which can be easily detected. However, the smart attackers would search for particular services with potential vulnerabilities. For example, if she looks for web servers, then TCP/443, TCP/80, TCP/8080 are targeted in priority and can reveal a service-semantic similarity. Similarly, an attacker may target a particular type of environment with various services close from a context-semantic point of view. For example, a web service relies usually on a web server and on database. So both of them are regularly co-located in a close network vicinity, even in the same host. In a previous work [4], this intuition has been confirmed.

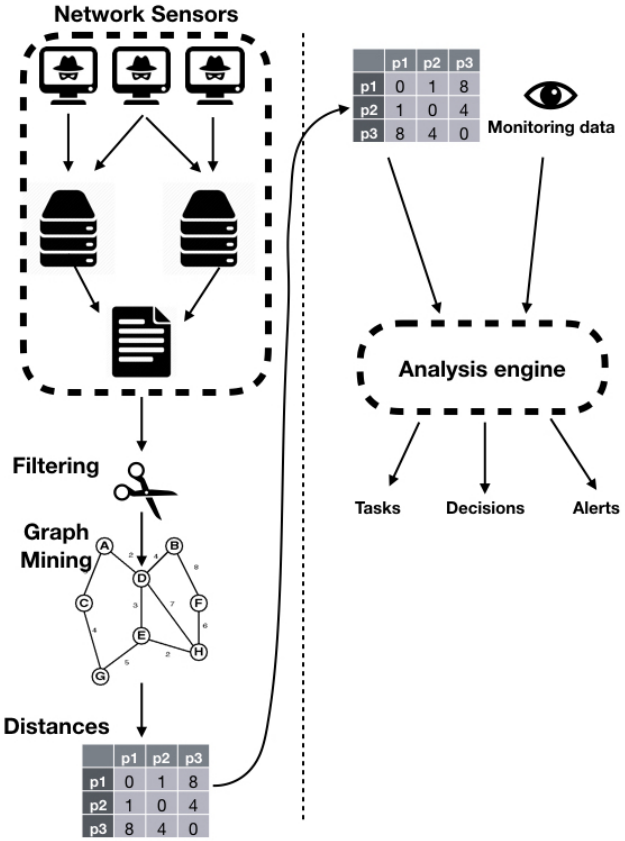


Fig. 1: Methodology overview

In this paper, we propose to aggregate, from the massive observation of a darknet, such a knowledge into a single similarity metric designed to catch both behaviors: the attackers looking for a particular type of service (service-semantic) and the attackers looking for services which are usually co-located together (context-semantic). This aggregation of the two semantics provides a smarter comparison for two given ports. We confirm the co-existence of these t¡o behaviors in scanning strategies of the attackers in section IV.

B. Methodology

Our aim is to extract the similarities between port numbers by observing attacker behaviors, *i.e.* ports targeted by the same attacker. More especially, we use a darknet or a network telescope that silently collects unsolicited traffic including TCP scans on a non active subnetwork. Indeed, an entire and never used IPv4 subnetwork acts as a black hole collecting all incoming packets, including some related to scans. Deriving a similarity from the distance between two ports targeted in a sequence does not lead to good results in preliminary experiments. Intuitively, a set of semantically close port numbers (either by context or by services) may not have been massively observed integrally (in a single sequence) but rather through multiple overlapping sequences.

All sequences must be aggregated together in a unique representation. In addition, even the ports that are supposed to

be similar are not targeted in the same order each. No global order should be constructed. We thus transform successively probed ports (sequences) by attackers into a unique graph.

Figure 1 illustrates the whole process to infer the similarity between port numbers:

1) Multiple attacker behaviors, e.g. scanned ports, are collected. In order to avoid a bias, it is required to collect such behaviors in a massive scale. In our case, we use a darknet or telescope (see section IV).
2) Scan extraction: since collected data can embed some noise, filtering is necessary and directly dependent on the collecting process. For example, big vertical scans running on all ports do not contain a valuable semantic and should then be discarded.
3) Graph building: the graph of scans is created from the filtered data. The nodes represent port numbers and the directed edge between two ports means that they have been probed sequentially at least once.
4) Similarity: a similarity measure between two port numbers is derived as the length of the shortest path between them.

While this method can be applied for both TCP and UDP, the remaining of the paper is focused on TCP ports. The main difference would be the scan extraction.

C. Extraction of network scans

Massive scan can happen on a very wide range of ports (sometimes all) that are not especially semantically connected (vertical scans). The same applies for horizontal scan targeting the same port (or a few of them) on numerous hosts.

Such scans are out of our interest to catch a supposed strategy in selecting ports by attackers. Our dataset is precisely described in section IV but Figure 2 represents the cumulative distribution of IP addresses per number of scanned TCP ports within a week. Our analysis is restricted to TCP SYN scans since they can be easily correctly isolated in our dataset. This figure highlights that most IP addresses (72%) are scanned with a limited number of port numbers during a week (less than 30). The long tail of the curve, here limited up to 75 (99.5%), represents so the vertical scan. A very wide number of IP addresses have less than 3 ports scanned, that is representative of a probing technique targeting few ports, *i.e.* horizontal scans.

Therefore, data is filtered according to these observations by discarding network traffic related to vertical (same IP address probed with more than 30 ports) and horizontal scans (an IP address probed with less than 3 ports) on a daily basis.

D. Graph-based port sequence model

The graph model is built from the method described in [4], that has highlighted semantic relationships between port numbers. The built graph represents all observed and filtered scans thanks to a well defined an summarized structure. A scan graph is a directed weighted graph $G = (N, E, \omega)$ with:

N The set of nodes of the graph. Each of them represents a unique TCP port.

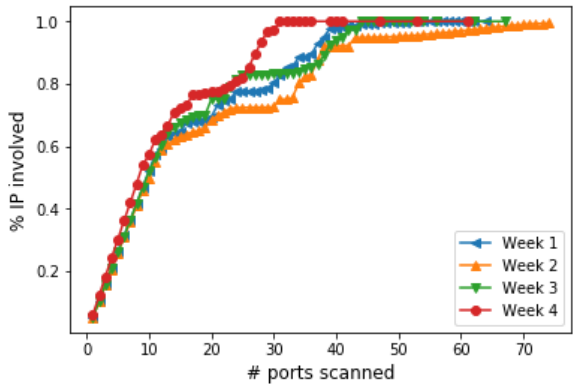


Fig. 2: Cumulative sum of number of ports scanned per destination IP address per week (in a 4-weeks period)

E The set of edges of the graph. An edge $e_{i,j}$ from port p_i to p_j exists if port p_j has been probed following p_i on the same destination IP address and from the the same source IP address (representing the source of the scan).

ω is a weight function for edges defined as $\omega(e_{i,j})$, the number of times p_j follows p_i in all scan sequences.

E. Shortest path based inter-port similarity

The defined graph contains the desired semantic. If two ports are connected by an edge with a high weight, they have been probed a lot of time successively. By generalization, the graph also contains ports that are near each other by transitivity. For example, if scans go repeatedly from 80 to 443 and from 443 to 3306 (MySQL), the graph contains a transitive link between ports 80 and 3306 and reveals thus a semantic similarity between these ports, but lower than between 80 and 443 (connected by a direct edge).

The intuition of this semantic is to swap (or invert) the weight of edges in the graph to reduce the shortest path length between ports which are regularly scanned in a same sequence (*i.e.* those connected together with heavy weights). Then, shortest paths $sp(n_i, n_j)$ between the pair of nodes n_i and n_j (port numbers) are computed. $sp(n_i, n_j)$ is the smallest sequence of edges from the source n_i to the destination n_j according to the inverted weights. The length $l(sp(n_i, n_j))$ of this shortest path is then used as a dissimilarity measure between the two ports, i and j, represented by the nodes n_i and n_j respectively. It is denoted as d_{sp}:

$$d_{sp}(i,j) = l(s) = \sum_{\forall e_{i,j} \in s} \omega'(e_{i,j}), s = sp(n_i, n_j) \quad (1)$$

Finding the shortest paths in a graph is a common problem. Methods, like the Dijkstra algorithm, are well defined. The main challenge resides in defining a correct rescaling and swapping method for the edge weights, *i.e.* deriving $\omega'(e_{i,j})$ from $\omega(e_{i,j})$ to make closer nodes linked with heavy weighted edges. In the original graph, the weight of an edge represents

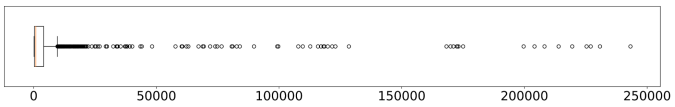


Fig. 3: Edges weights boxplot

the number of times a transition occurs between two ports. For the sake of clarity, we will simplify $\omega(e_{i,j})$ weights notation to $\omega_{i,j}$.

The distribution of edges weight, θ, in the graph is given in Figure 3 based on our dataset described in section IV. In this figure, we can observe an unbalanced distribution with most of the values concentrated between $Q_1 = 299$ and $Q_3 = 4082$ (θ inter-quartiles range - IQR). Therefore, data needs to be rescaled. Due to the occurrence of outliers, we have chosen to use the IQR as a basis for rescaling as follows:

$$\omega_{i,j}^{iqr} = \frac{\omega_{i,j} - Q_1(\theta)}{Q_3(\theta) - Q_1(\theta)}, \theta = \{\omega_{i,j} \forall i, j\}$$

Actually, the IQR-based rescaling method relies on data distribution instead of values and spreads the most represented values on a larger range. Because rescaled values originally below Q_1 become negative, we shift the rescaled data to positive values by deducing the minimal value, $\omega_{i,j}^{iqr} - \lambda$ with $\lambda = \min_{i,j}(\{\omega_{i,j}^{iqr}\})$.

Finally, weights can be swapped regarding the maximum value (which is also shifted):

$$\omega'_{i,j} = (\max_{i,j}(\{\omega_{i,j}^{iqr}\}) - \lambda) - (\omega_{i,j}^{iqr} - \lambda)$$

This data-driven scaling and swapping technique avoids to use arbitrary factor when inverting the edge weights.

IV. Darknet overview

A darknet also known network telescope or Internet blackhole is an entire reachable subnetwork collecting all incoming traffic with no active hosts. It has been proved to contain valuable information to understand major security threats like DDoS attacks and scanning activities [16].

A. Datasets and pre-processing

Two darknets are used in this paper over 40 weeks of observations. The first one (FR), in France with a dedicated /20 subnetwork. The second one (JP) is a /20 subnetwork in Japan. Using both datasets strengthens our evaluation, especially to assess if there are dependencies between locations.

General statistics are provided in Table I. The Japanese darknet attracts more traffic than in France one but from less attackers meaning that people attacking Japanese darknet use significantly more packets in their scan probes.

In next sections, detailed statistics about the observed port scans are provided to understand the attacker behaviors. Except when mentioned, all statistics given in the next sections are presented over a joint dataset including both the JP and FR datasets.

TABLE I: General darknet statistics (attackers are identified by unique source IP addresses)

	France	**Japan**
Begin date	1st January 2015	1st January 2015
End date	30th September 2015	30th September 2015
Total # of attackers	3,771,092	3,712,209
Average # of attackers per day	19,776.66	19,621.43
Total # of packets	399,344,813	415,642,444
Average # of packets per day	1,426,231.47	1,484,437.3

B. Number of scans

Figure 4 shows no explicit correlation between the day of the week and the number of scans. However, we can notice that the number of scans a day is always between around 9 million and 17 million. Moreover, on Saturday, less variations are observed.

In Figure 5, we evaluate the number of times an attacker (identified by the source IP address) targets the same port on the same destination IP address within the same day. Such a value is actually very high. Once a scan detected, an efficient pre-emptive blocking technique should always block the associated port as it will be undoubtedly targeted again by the attacker. Actually, most of scanning repeats the same probing packet to increase the validity of the reply.

C. Number of distinct targeted ports in scans

Our datasets reveal that all TCP ports are targeted within a week. In fact, even a single vertical scans can lead to such a situation. With a more fine-grained focus, in Figure 6, the curve entitled *distinct destination ports* depicts the average number of targeted ports per destination IP address.

This number is around 350 that is relatively low compared to the to the 65.536 available TCP ports. More precisely, 90% of IP addresses are scanned on less than 900 ports a week (with a mean around 400 ports) as highlighted in the same figure. Hence, the scans target selected ports and the probing strategy is not random, that validates our main assumption for our work (attackers behavior is not random and semantic can be extracted from probing activities). Moreover, Figure 6 also presents the average number of new ports scanned per destination IP address each week. Compared to the previous curve, the dynamic is the same with a very similar shift along the weeks. There is so a similar number of new port numbers targeted every week.

Intuitively, the targets of the attacks may be motivated by the apparition of newly discovered vulnerabilities in devices and services. Our observation confirms this intuition and quantifies it. Besides, modeling the attacker behavior has to be done over long periods and need to be reassessed regularly in order to update the graph that serves as inferring the similarity metric between ports. This would avoid to catch ephemeral behaviors.

D. Inter-scan time

Another question regarding the scanning behavior is the vivacity of a TCP port scan denoted as the inter-scan time. It

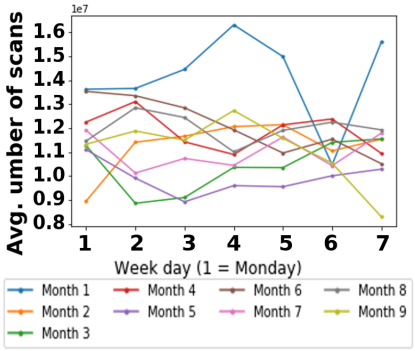


Fig. 4: Average number of rescan

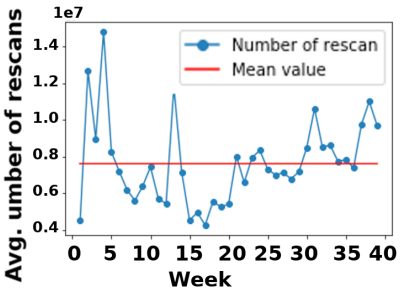


Fig. 5: Number of re-scan in a day by week.

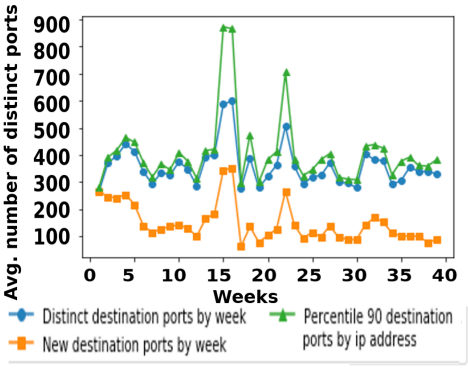


Fig. 6: Average number of ports scanned by IP address

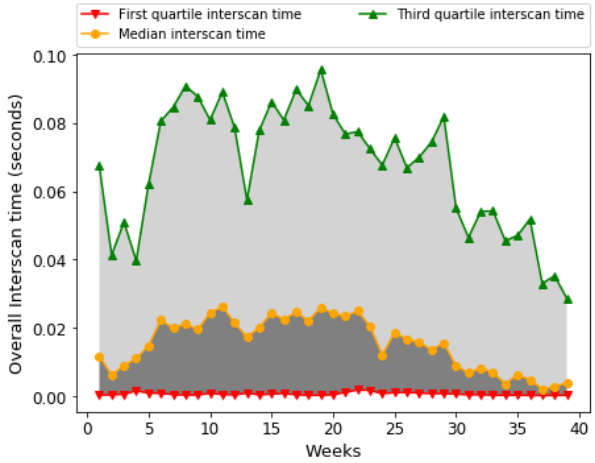


Fig. 7: IQR distribution for overall inter-scan time

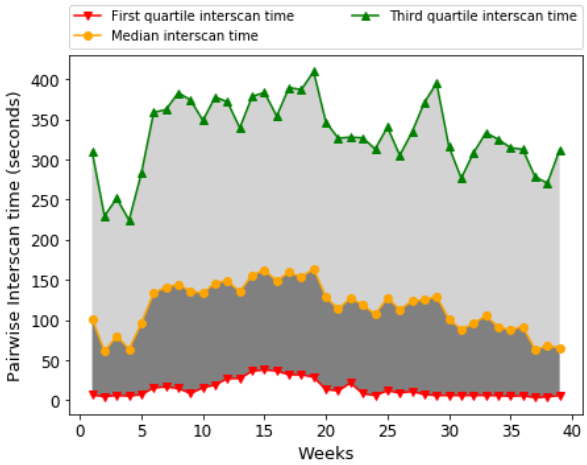


Fig. 8: IQR distribution for pairwise (source and destination IP address) inter-scan time

is the average elapsed time between two consecutive probed ports in the darknet. This gives an insight revealing if attackers prefer to use long stealthy or short scans. It also helps to fine-tune our proactive scan blocking technique (in Section VI) in regards to the period of time an IP address should be filtered.

In Figure 7, we compute the inter-scan time over the FR and JP dataset on a weekly basis. The median is around 10ms, so the darknet receives an average of 100 scans probes by seconds but the frequency tends to increase over the weeks. Proactive blocking is helpful to reduce the footprint of network scans and also by nature blocks the attacker in gathering information for crafting future attacks.

The inter-scan is computed considering both source and destination IP addresses (pair-wise) before being averaged in Figure 8. The goal is to isolate the behavior of a single scan. In this case, the median time is around 100 seconds between the scans.

In a nutshell, we conclude that the scans are constantly observed with an increasing frequency over the months as we expected. Furthermore, in many cases, a single source IP address targets few ports with some delays in probes in order to evade detection techniques.

V. Evaluation

In order to assess the veracity of our proposed similarity and because no ground truth actually exists, we first extract the smallest dissimilarities, which are thus representative of the most semantically-linked port numbers.

We only represent the 60 smallest values as annotated edges between ports in Figure 9. We can distinguish the smallest dissimilarities (in bold), and so higher similarities, between HTTP-related ports: 80 (HTTP), 443 (HTTPS) and 8080 (Alternative HTTP). Moreover, these ports are also connected to email service ports. This is also relevant because an email server is frequently used by web services for instance (to send notifications). FTP port is also very close to web ports. Indeed, FTP was largely used in the past for updating web pages (especially personal home pages). Other relations like between ports 22 and 3389 are logic because they are related to standard services (SSH and remote desktop) to open session on remote computers. This shows the ability of our semantic similarity to extract and represent several types of semantics between network ports.

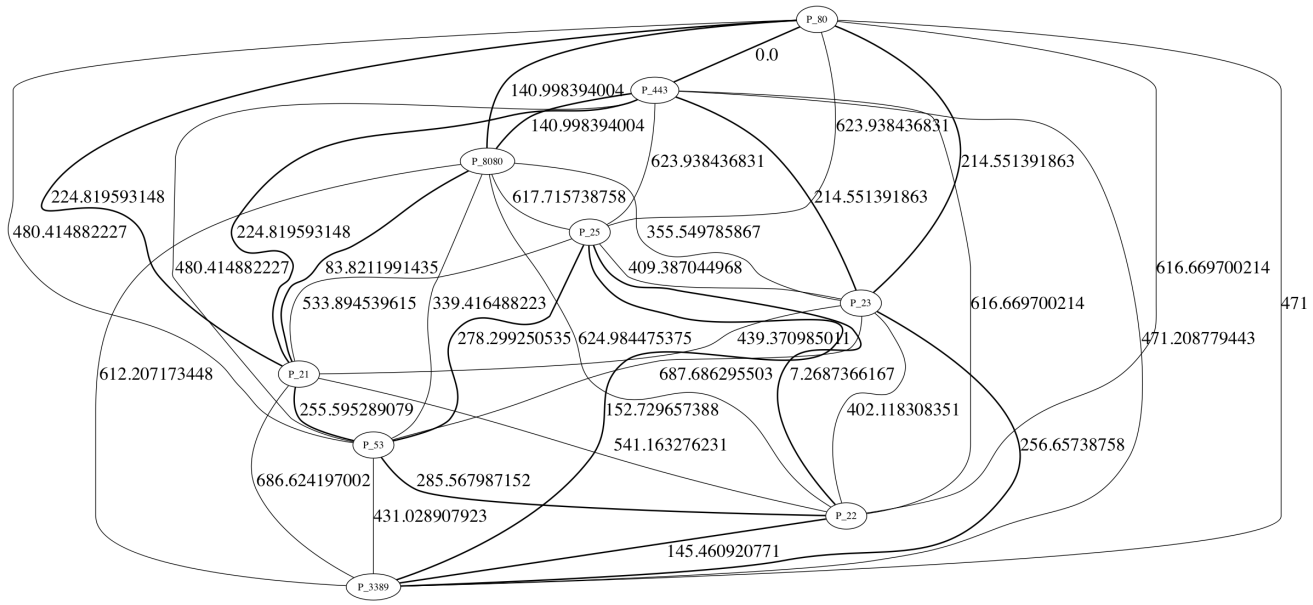


Fig. 9: Network ports graph linked with the 60 smallest *shortest-path-based* similarity

VI. Preventive port blocking

In this section, we present an application of the attacker behavior-based similarity to demonstrate its validity under a realistic context. We rely on a single use case serving as an example of application of our proposed metric. This one have been selected because we have ground truth to perform the evaluation.

Assuming that a probed port is detected with a regular method such as those deployed in intrusion detection system (*e.g.* based on the number or the ratio of TCP connection requests to closed ports), our preventive port blocking aims to predict port numbers that will be probed next to discard the traffic accordingly in advance. It may be thus a part of an Intrusion Prevention System (IPS). It is worth noting that existing detection methods are able to detect scans after 4 or 5 attempts [20].

A very restrictive technique could fully blacklist an IP address performing a scan but our approach is more fine-grained by blacklisting selected ports only. This avoids collateral effects when an IP address is shared by multiple users, for instance with NAT. Depending where such a system would be deployed, it can be applied against private or public IP addresses (egress vs. ingress filtering for instance). Because probing or scanning is an initial step to discover reachable hosts and services, defeating it reduces the attacker visibility and so limits her ability to craft a very tailored attack.

Assuming an IP address is detected as performing a scan towards a given port number, our method pro-actively blocks the ingoing traffic from this IP address towards the K nearest ports for a user-defined period of time. For instance, when a scan targets port 80 (HTTP) our method filters traffic towards 443 and 8080 assuming $K = 2$ (as well as the initial port). The method is voluntary simple (compared to sophisticated methods with advanced modeling or techniques like machine learning) in order to focus our evaluation on the veracity of our new inter-port similarity. The advantage is also to limit the overhead because the similarities are computed beforehand.

A. Evaluation methodology

The similarity between ports are derived using the overall dataset (combining both the JP and FR darknet) between January and June 2015 while the period from 1st to 7th July 2015 is used for testing. We define two performance metrics:

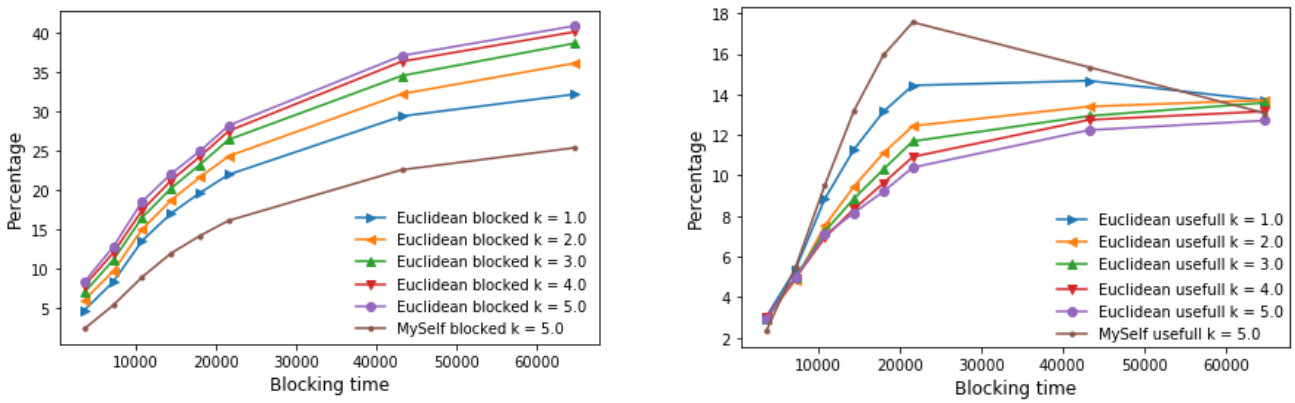
1) Blocking ratio: The percentage of probed ports pro-actively blocked (%blocked).
2) Usefulness: The percentage of blocked ports which are effectively probed afterwards (%usefulness).

In fact, quantifying the number of false positives is impossible because the darknet data does not contain mix traffic including legitimate traffic. Our usefulness metric is thus more drastic by only considering as valid, only blocked ports observed then in the next scans. However, in section VI-D, a real dataset with mix traffic is used in order to evaluate the number of false positives.

B. Baseline scenarios

In order to assess the benefit of our metric, two baseline scenarios are considered:

- *MySelf* consists in solely blocking the current probed port number since our observations in section IV shows that an attacker usually targets the same ports multiple times.
- *Euclidean*: this algorithm blocks naively the K nearest ports using an euclidean in addition to the initially probed port. If the scanned port is 80, and K = 2, the set of ports to block will be 80, 81 and 82 (when equality, the highest port number is selected in priority, *e.g.* 81 with $K = 1$)



(a) Blocking performance

(b) Usefulness performance

Fig. 10: Port blocking - baseline scenarios

We also vary the blocking time in Figures 10(a) and 10(b). The *MySelf* strategy leads to block between 5 and 25% of scans with a usefulness always lower than 18%.

In Figure 10(a), more scans are pro-actively blocked with the *Euclidean* distance algorithm though the usefulness, in Figure 10(b), is lowered and so potential false positives could increase in a real case of mixed traffic.

Regarding the parameters, increasing K leads to increase the number of blocked ports including those which are effectively probed afterwards and those which are not aimed by the attacker. As a result, the blocking ratio logically increases. However, as highlighted by the decreasing the usefulness, new ports blocked when increasing K have a higher probability to not be targeted in the future. Hence, increasing K is not a good strategy to improve the efficiency of the baseline Euclidean-based technique.

A higher blocking time also contribute to block more ports. It tends to enhance the usefulness until an upper-bound around 20000 seconds (5h30). Therefore, the port blocking is efficient with this time horizon.

C. Results

In this section, we assess the benefit of the proactive blocking using our defined similarity compared to the baseline scenarios considering the impact of two main parameters: (1) the number of ports to be blocked (K) and (2) the time the ports (for the considered IP address) are blocked. We compare results assuming JP or FR datasets, in Figure 11(c) and Figure 11(b) respectively, or both together in Figure 11(a) (similarly to the baseline scenarios).

There is a significant improvement for the blocking percentage and the usefulness. Up to 40% and 30% of scans are blocked for the FR and JP dataset respectively in Figure 10. Globally, the blocking ratio can reach around 70% in Figure 11(a) ($\sim \times 3$ increase). Besides, the percentage of usefulness is about 15% for FR and near 12.5% for JP giving a maximum global usefulness of 30% compared tothe baseline scenarios with 18%.

Regarding the impact of parameter values, the usefulness increases when the blocking period of time increases until around 5h30 with the overall dataset. Unlike baseline scenarios, increasing K may be beneficial for the two performance metrics. Actually, a good trade-off between blocking ratio and usefulness is $K = 3$ in order to block around 50% of all scans with a usefulness of 25% assuming an optimal blocking time of 20000 seconds.

D. Test with real traffic

Based on similarities learnt on the darknet data (both JP and FR) and the best tuning of parameters highlighted in the previous section, the proactive blocking technique has been applied to the MAWI Labs dataset [22].

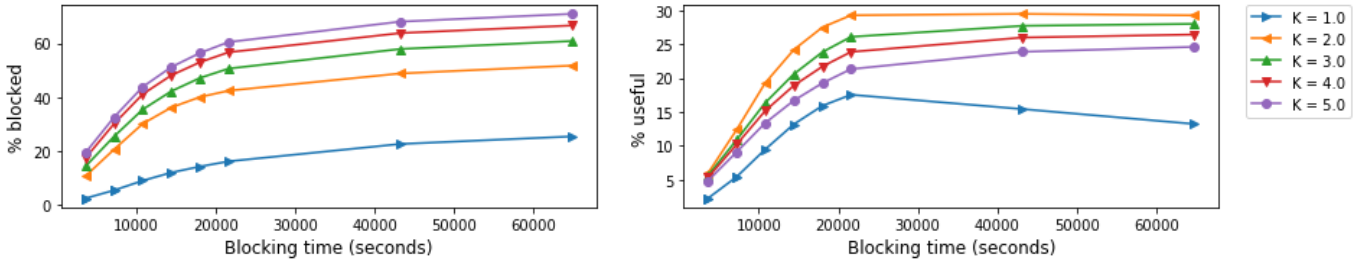
It contains real, and so also benign, traffic captured from an oceanic backbone between United States and Japan. Therefore, the false positive rate (FPR) can be calculated in order to check if the preventive blocking (using our new inter-port dissimilarity metric) does not impact benign traffic by discarding ports which are used by the latter.

Having a low usefulness (always lower than 30% in our previous experiments) may not be a problem if predicted ports, and so blocked ports, are not used by the legitimated communications either. In that case, the FPR remains low. However, if the automatically blocked port affects benign traffic, the FPR increases.

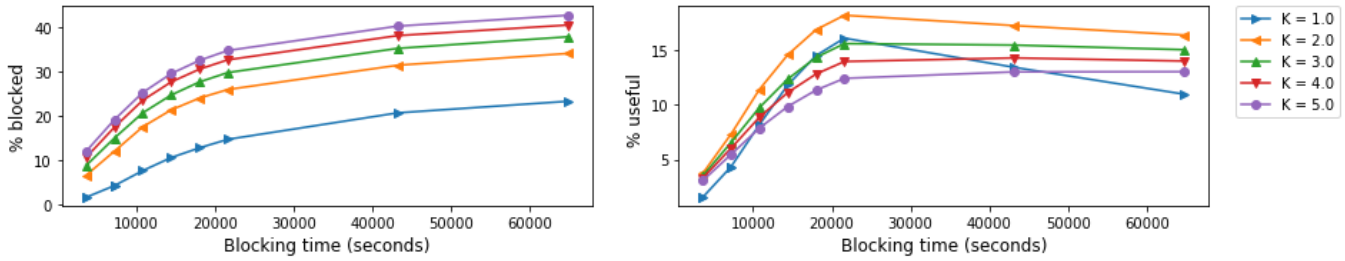
We consider the period from 2 to 9 September 2015 (except the 5th and 7th of September because of dataset unavailability) with a total of 590,173,645 IP packets with a mean of 98,362,274 packets a day. Each day is composed of 15 minutes.

Because only 15 minutes of data per day are labeled, a 5h30 our blocking time is not relevant and is so set to 15 seconds.

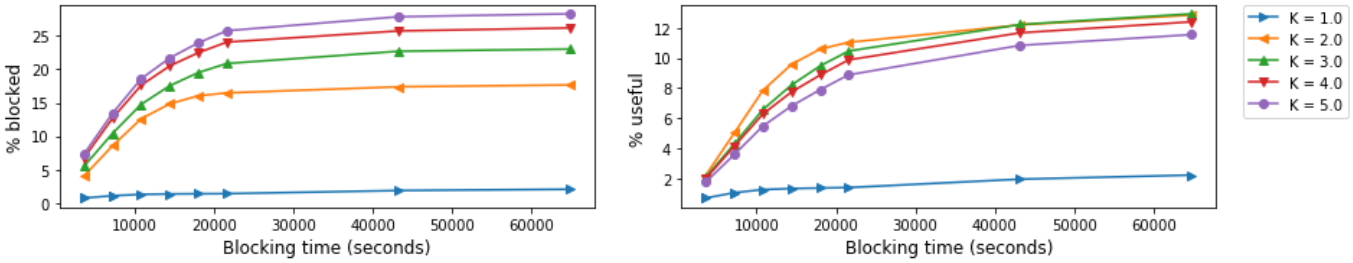
As shown in Table II, 99.9% of scans are effectively blocked in a proactive manner. It is due to lower variety in the targeted ports compared to what is collected by a dedicated security sensor such as the darknets. The usefulness presents also higher values but the most interesting is the low FPR largely under 0.01% decreasing to near 0%. These results



(a) Results for both darknets (FR, JP)



(b) Results for french darknet



(c) Results for japanese darknet

Fig. 11: Blocking statistics for the K-Nearest ports with the attacker behavior-based similarity

prove that the proposed semantic port similarity, derived from the knowledge generated by the observed attacks, can cleverly block network ports before an attack or a network scan occur. Furthermore, this experiment has shown that knowledge inferred from our darknet is not biased by a particular context or location as we used it in another environment.

VII. Conclusion

In this paper, a new attacker behavior-based inter-port measure is introduced. The metric we introduced extracts an attackers' behavioral model, from real scanning activities, which embeds an underlying semantic about the targeted ports. The observations done with a darknet over a long time period motivate the definition of this new measure between port numbers. In order to assess its viability in an extensive manner, a proactive blocking technique has been defined and tested.

	TPR	FPR	Usefulness
Minimum	99.94%	0.0000012%	47.79%
Mean	99.98%	0.0015%	66.97%
Maximum	99.99%	0.0091%	83.33%

TABLE II: Preventive port blocking in real network with K=3

Using real world data, we showed that more than 99% of scans can be blocked in advance with less than 0.1% of legitimate traffic blocked. The latter proves that the knowledge extracted from our darknet observations contains rich information to derive an inter-port similarity measure, which is robust enough to be applied in another context (in a different network).

Similarities between ports are daily updated and publicly accessible at http://port2dist.lhs.inria.fr/.

Future work will refine or extend our proposed metric by including other sources of information (such as RFCs) to determine inter-port similarities.

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) providing the French darknet data and we gratefully thank its technical director, Frederic Beck from Inria Nancy Grand Est. We also thank the NICT providing the Japan darknet data (https://www.nicter.jp).

REFERENCES

[1] M.-S. Kim, H.-J. Kong, S.-C. Hong, S.-H. Chung, and J. W. Hong, "A flow-based method for abnormal network traffic detection," in *Network Operations and Management Symposium (NOMS)*. IFIP/IEEE, 2004.

[2] M. Sheikhan and Z. Jadidi, "Flow-based anomaly detection in high-speed links using modified gsa-optimized neural network," *Neural Computing and Applications*, vol. 24, no. 3, pp. 599–611, Mar 2014. [Online]. Available: https://doi.org/10.1007/s00521-012-1263-0

[3] 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

[4] S. Lagraa and J. François, "Knowledge discovery of port scans from darknet," in *Symposium on Integrated Network and Service Management (IM)*. IFIP/IEEE, 2017.

[5] 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.

[6] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Passive and Active Network Measurement*, C. Dovrolis, Ed. Springer, 2005.

[7] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355–374, 2015.

[8] A. K. J. Michael, E. Valla, N. S. Neggatu, and A. W. Moore, "Network traffic classification via neural networks," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-912, Sep. 2017. [Online]. Available: http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-912.pdf

[9] W. D. Donato, A. Pescape, and A. Dainotti, "Traffic identification engine: an open platform for traffic classification," *Network*, vol. 28, no. 2, pp. 56–64, March 2014.

[10] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust network traffic classification," *Transactions on Networking*, vol. 23, no. 4, Aug. 2015.

[11] 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.

[12] 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.

[13] M. H. Bhuyan, K. Bhattacharyya, and J. K. Kalita, "Surveying port scans and their detection methodologies," in *The Computer Journal*, vol. 54, 10 2011, pp. 1565–1581.

[14] C. B. Lee, C. Roedel, and E. Silenok, "Detection and characterization of port scan attacks."

[15] M. Coudriau, A. Lahmadi, and J. Francois, "Topological Analysis and Visualisation of Network Monitoring Data: Darknet case study," in *International Workshop on Information Forensics and Security (WIFS)*. Abu Dhabi, United Arab Emirates: IEEE, 2016. [Online]. Available: https://hal.inria.fr/hal-01403950

[16] C. Fachkha and M. Debbabi, "Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization," *Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1197–1227, 2016.

[17] E. Balkanli, J. Alves, and A. N. Zincir-Heywood, "Supervised learning to detect ddos attacks," in *Computational Intelligence in Cyber Security (CICS), 2014 IEEE Symposium on*, Dec 2014, pp. 1–8.

[18] P. Mell and R. Harang, "Limitations to threshold random walk scan detection and mitigating enhancements," in *Communications and Network Security (CNS)*. IEEE, 10 2013, pp. 332 – 340.

[19] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan, "Fast portscan detection using sequential hypothesis testing," in *Security and Privacy*. iEEE, 5 2004, pp. 211 – 225.

[20] ——, "Fast portscan detection using sequential hypothesis testing," in *Security and Privacy*. IEEE, 5 2004, pp. 211 – 225.

[21] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Communications and Multimedia Security*. Springer, 2014.

[22] (2018, 9). [Online]. Available: http://www.fukuda-lab.org/mawilab/v1.1/index.html