On the Impact of Flow Monitoring Configuration

Petr Velan, Tomas Jirsik
Institute of Computer Science
Masaryk University
Brno, Czech Republic
{velan,jirsik}@ics.muni.cz

***Abstract*—Flow monitoring has become an essential source of information for intrusion detection systems and various forms of network data analytics. However, the attention of researchers is focused primarily on the utilisation of the flow data, and the process of flow data creation is often neglected. This lack of consideration negatively affects the results of data analytics. Either the results are suboptimal due to the low quality of the flow data, or a description of the configuration of the flow monitoring system is missing, which leads to irreproducible results. The goal of this paper is to demonstrate how the configuration of the flow monitoring system affects the resulting data. The most basic flow monitoring configuration variables are the flow expiration timeouts. We analyse their effect on the number of created flow records to show their importance. Moreover, we demonstrate that the choice of the flow expiration timeouts can have a severe impact on the network data analytics. The use-case of Slowloris attack detection is used as an example to illustrate this fact.**

***Index Terms*—network, flow, expiration, timeout**

I. Introduction

Machine learning has become a popular and widely used tool for network traffic classification and attack detection. To utilise this technique, appropriate features must be extracted from the network traffic, either directly from each captured packet or from aggregated flow records [1]. The usage of flow features is dominant, especially when a large volume of traffic needs to be processed since the flow monitoring is designed for large and high-speed networks. However, the generated flow features are affected by the settings and implementation of the flow monitoring system, which in turn affects the output of the machine learning algorithms. Therefore, it is necessary to study and address the impact of flow monitoring configuration on the resulting flow records.

The most important parameters of a configuration of a flow monitoring system that affects the creation of flow records are the flow expiration conditions [2]. These conditions directly influence how a flow is described by flow records. For example, for a ten-minute connection, one-minute active timeout splits the flow into ten different flow records. Setting the active timeout to five minutes results in only two flow records. Therefore, features extracted from these flow records, such as a number of packets, bytes or duration, will differ significantly. However, researchers often omit the description of the flow monitoring configuration from their research, which hinders the reproducibility of presented experiments.

978-1-7281-4973-8/20/$31.00

The goal of this paper is to show that the configuration of the flow monitoring system significantly affects the measurement results and should always be included whenever flow records are used in research. To achieve this goal, we execute flow monitoring with different configurations on two publicly available datasets and analyse the resulting flow records. We show how the number of generated flow records differs based on the flow expiration conditions. Furthermore, on the example of a Slowloris attack, we show that the machine learning attack detection is heavily influenced by the flow monitoring configuration.

The contribution of this paper is three-fold:

- An study of the impact of flow expiration timeouts performed on two public data sets. This study focuses on the difference in the number of created flow records.
- An analysis of the impact of flow expiration timeouts on subsequent data analysis. We argue that anomaly and attack detection methods are likely to be seriously affected by a change in flow monitoring configuration. This fact is demonstrated on a dataset capturing a Slowloris attack.
- We propose guidelines for inferring flow expiration timeouts based on the expected traffic characteristic and network operation criteria.

The rest of the paper is organised as follows. Section II outlines the researched problem and provides necessary background information. Section III discusses related work. Section IV describes the used datasets, our approach to data analysis, and briefly mentions the implementation of the tool used for the data analysis. Section V examines and interprets the results of the performed data analysis. Section VI discusses the impact of the results on the flow monitoring system configuration. Moreover, it provides guidelines for determining appropriate flow expiration timeouts. Section VII concludes the paper.

II. Problem Statement

Using the correct configuration for the flow monitoring system is essential since different configurations result in different flow records for the same traffic. Using different configuration or withholding the configuration altogether has several undesirable impacts on the further processing of the data. Firstly, without the knowledge of the proper configuration, experiments using flow monitoring infrastructure are not reproducible. Secondly, shorter flow records (i.e. created using lower timeouts) carry more information than the shorter

ones. In the extreme cases, creating a flow record per each packet maintains the highest informational value and creating a single flow record for each connection carries the least information. Therefore, analysis of the flow records, especially using statistical and machine learning techniques, can be severely affected by the choice of flow monitoring configuration. Lastly, processing of generated flow records, especially using sophisticated data analysis algorithms, can be computationally expensive. Therefore, creating unnecessarily many flow records can overwhelm the flow analysis engine. This section describes the flow creation process with a focus on flow expiration conditions, as they are easily configured and affect the flow creation process the most.

The flow creation process can be divided into three basic parts: packet metadata extraction, flow aggregation, and flow export. This paper focuses on flow aggregation and especially the influence of flow expiration conditions on flow aggregation. When a packet is captured by the flow monitoring system, the metadata extraction process retrieves information that is used to create new or update an existing flow record. This process is called flow aggregation. If a matching flow record does not exist, it is created using the packet metadata. Otherwise, an existing flow record is checked for expiration, and if it is still valid, it is updated. Moreover, the existing flow records need to be periodically checked for inactivity so that they can be exported in a timely manner.

The RFC 5470 [2] defines three flow expiration conditions: *inactive timeout*, *active timeout*, and *resource constraints*. In practice, there are at least two other reasons for flow expiration, *end of flow* and *exported shutdown*. All of these flow expiration conditions are accounted for in RFC 5102 [3] and can be indicated by the *flowEndReason* Information element. This paper focuses on the impact of active and inactive timeouts on the generated flow records.

Inactive timeout is a primary expiration mechanism which ensures that the flow records leave the flow cache at all. Too large inactive timeout causes the flow records to be kept in the cache longer, which causes higher resource consumption and delays processing of these flow records. Short inactive timeout splits the flows with large interpacket gaps unnecessarily.

Active timeout causes expiration of long-running flows on a regular basis. The active timeout ensures that the connection is observed after this timeout at the latest. Otherwise, keeping a connection open for a long time would prevent its corresponding flow record being observed and processed in a timely manner, which could impact accounting and security.

III. Related Work

One of the first works to analyse the impact of timeouts on the creation of flows is [4] by Claffy et al. The authors use a different definition of flows than this paper and consider only a single timeout. They study the packet volumes, byte volumes, and flow duration for timeouts ranging from 2 to 2048 by powers of two. The results show that lower timeouts require keeping fewer flow records in the cache, but a higher number of new flows must be created per time unit since more flows are prematurely evicted from the cache. The authors also analyse the impact of keeping and creating the records has on the performance of routers and suggest optimising the timeouts to minimise these costs.

The authors of [5] analyse artefacts of routers exporting J-Flow. They use synthetic packet samples with different packet interarrival times and measure the number of flows generated by the devices depending on inactive flow expiration timeout. The results show that one implementation of the flow monitoring systems ignore the inactive timeout altogether while the precision of the other one varies based on the size of the inactive timeout. In this paper, we assume that the flow monitoring system behaves exactly as specified and does not skew the results in any way.

The work of Hofstede et al. [6] focuses on flow measurement artefacts as well. One of the analysed artefacts is imprecise flow record expiration. The authors used synthetic traces as well to test the precision of active and inactive timeout of six flow exporters under different configurations. Most of these exporters behaved inconsistently and split the flows differently than expected. This resulted in a different number of generated flows, which can affect the results of further processing, as explained in the previous section.

Rodríguez et al. consider different traffic classes for establishing optimal flow timeout in [7]. Data traces from ISP and mobile operator are classified using DPI tool PACE. Each traffic category is analysed separately to determine packet interarrival times and derive appropriate timeouts. The authors do not focus on flow monitoring and are interested only in the inactive timeout. The results show that by optimising the timeout, significant reduction of the number of flows and consequently used memory during the data processing can be achieved.

IV. Methodology

The goal of this paper is to study the impact of flow expiration timeouts on the created flows. We have used existing, publicly available datasets for our analysis. To use existing flow exporters on these datasets would require rigorously testing the correctness of these exporters. Moreover, we only need a small subset of features provided by flow exporters, i.e. to identify packets belonging to different connections and then split those connections into flow records based on the active and inactive timeouts. Therefore, we created our own tool for this specific task and released it for public use.

The rest of this section describes the used datasets, performed data analysis, and implementation of the algorithm for computing flow records.

A. Datasets

Two different datasets are used in this paper. The first is *The CAIDA Anonymized Internet Traces 2015 Dataset* [8]. It contains an hour of high-speed backbone traffic captured on 21 May 2015. Both directions of the traffic are stored separately and every minute of the capture is stored in a separate compressed pcap file. We have decompressed the files

TABLE I
DESCRIPTION OF USED DATASETS.

	Dataset1	Dataset2	Dataset3
File size	82.5 GB	1.3 GB	20.1 MB
TCP packets	987,143,650	3,564,110	106,933
TCP connections	14,333,540	438,783	12,263
UDP packets	126,733,857	281,691	0
UDP connections	7,111,079	77,445	0
ICMP packets	1,243,588	5,567	0
ICMP connections	325,638	630	0
Capture length	1:02:03	33:31:34	0:41:48

on a direction A of the traffic and merged all the one-minute pcap files into a single large pcap file. The resulting 82.5 GB pcap file is used in our analysis and will be called Dataset1 in the rest of this paper.

The second dataset used in this paper is *A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)* [9]. This dataset contains records of multiple different network attacks carried on in a laboratory environment. There are pcap files and logs captured during the attacks, and the whole dataset is split into ten days. For the purpose of this paper, we used two subsets of the whole sample. The first subset (called Dataset2 from now on) comprises of the whole first day, 3 February 2018. The second subset (called Dataset3) comes from 15 February 2018 and contains only the captured Slowloris attack, filtered by IP address range provided in the description of the CSE-CIC-IDS2018 dataset. Both subsets were created as a (time-ordered) merge of multiple pcap files that are present in the original dataset.

Table I describes the basic properties of each of the used datasets.

B. Data Analysis

TCP, UDP, and ICMP protocols are the most common transport protocols found in the network traffic. Since their purpose and behaviour is different, we have decided to study them separately for each dataset. Therefore, each dataset was split into three parts by the transport protocol. After that, we grouped packets belonging to distinct connections so that we could easily compute flows with different flow expiration timeouts for these connections.

We have computed the following features of the datasets (each feature for all transport protocols):

- Number of connections.
- Distribution of connection lengths.
- Distribution of interpacket gaps (per connection).
- Number of flow records for various expiration timeouts settings.

The number of connections gives a lower bound on the number of flow records that are created for the given dataset. The number of packets is the upper bound, but the timeouts would have to be close to zero to split the connections into that many flow records. The distribution of connection lengths helps to show how the active timeout influences the number of generated flow records.

TABLE II
CONNECTION LENGTH DESCRIPTIVE STATISTICS (IN SECONDS).

Data	Proto	Mean	25%	50%	75%	Max
1	TCP	85.47	0.26	1.05	12.83	3722.98
	UDP	228.35	0.00	0.00	0.00	3722.98
	ICMP	608.09	0.00	0.00	488.69	3722.71
2	TCP	412.01	0.01	0.01	1.37	31910.15
	UDP	1753.76	0.00	0.00	0.02	118953.09
	ICMP	2965.13	0.00	0.00	7.80	31272.45
3	TCP	194.38	105.76	106.94	107.68	2481.99

The inactive timeout is needed to ensure that the flow records are exported at some point in time when no new packets of the connection are arriving. Ideally, it would be applied only after the end of the connection and would not influence the number of flow records. However, to ensure that the information is exported with a fixed delay and also to free the used resources (especially the memory taken by the record of the connection), it is being set to a value that is lower than some of the interpacket gaps in the connections. The distribution of interpacket gaps helps to explain the relation between inactive timeout and number of generated flow records.

To compute the number of flows for various expiration timeouts, we simply run the algorithm which splits the connections to flow records with various settings. We consider timeouts from 1 to 600 seconds (with an increment of one second) to include very short timeouts that save resources and long timeouts that avoid unnecessarily splitting the connections. Active and inactive timeouts are computed separately (without the other one being used) and in all combinations where the inactive timeout is less than or equal to the active timeout. The only exception is the Dataset1, where the step for the combination of timeouts is two seconds due to the complexity and length of the computation. We have made the tools used to compute flow records available on Github [10].

V. RESULTS

This section shows the results of the analyses described in the previous section. First, we present the dataset characteristics that shed light on the possible impacts of the flow timeout settings. Next, we discuss the impact of both active and inactive timeout settings individually. After that, we provide the results of the joint impact of both timeouts on the flow measurement. Last, we demonstrate how different settings of the flow expiration timeouts affect the detection methods on the Slowloris attack use case.

The connection counts for individual protocols (see Table I) reflect the number of the packet presented in the datasets. We can observe that the TCP and UDP connections aggregate more packets than ICMP connections. The reason is that ICMP is a service protocol and is not designed to transport data.

Table II describes the distribution of the length of the individual connections present in the datasets. The minimum

TABLE III
INTERPACKET GAPS DESCRIPTIVE STATISTICS (IN SECONDS).

Data	Proto	Mean	25%	50%	75%	Max
1	TCP	0.26	0.00	0.00	0.01	3717.41
	UDP	13.58	0.00	0.01	0.02	3720.85
	ICMP	215.72	1.02	18.70	306.94	3714.40
2	TCP	57.845	0.00	0.01	0.32	29844.53
	UDP	664.98	0.00	0.00	2.02	83134.49
	ICMP	378.37	7.20	23.33	50.14	25159.65
3	TCP	25.18	0.42	3.41	31.00	250.19

values for all datasets and protocols were equal to zero and are omitted from the table. We observe that connection length distribution is skewed as most of the connections are very short. However, the mean values indicate that there is a small number of very long connections as well. ICMP connections are longer than TCP and UDP connections. The reason is that they are not differentiated by transport ports, and therefore any ICMP communication between two hosts is part of the same connection.

Dataset3 shows different distributions in the connection lengths than the other two datasets since it contains only traffic of the Slowloris attack. The reasons for the specific connection lengths are discussed in Subsection V-D.

The interpacket gap distribution analysis is presented in Table III. We observe that it is skewed similarly to the connection length distribution, i.e. there is a small number of large interpacket gaps. For example, given the fact that the inactive timeout is usually set to 30 seconds, setting the inactive timeout to different values will influence the properties of more than 50% of the connections in the Dataset1.

A. Impact of the Inactive Timeout

We have argued that the impact of inactive timeout depends upon the interpacket gaps of the observed connections. Figure 1 represents connections of the TCP protocol of the Dataset1. Only inactive timeout is applied, the active timeout is not used in this figure. The x-axis represents different inactive timeouts in relation to the number of flows (the purple line). It also represents the length of interpacket gaps in their histogram (blue bar graph). We can clearly see that the number of flows correlates with the interpacket gaps. For example, since there is a distinct number of interpacket gaps approximately 45 seconds long, it causes the number of connections to decrease when the inactive timeout is set to a higher value. The same dependency can be observed at other values as well.

To verify that our assumption is correct, we computed the number of interpacket gaps longer than 45 seconds and equal or shorter 46 seconds. It is exactly the same as the difference between the number of flows for the 45 and 46-second inactive timeout.

Looking at the histogram of interpacket gaps can help us determine which value to use for an inactive timeout. Values just after large spikes in the graph should be preferred as they avoid unnecessary splitting of connections into multiple flow records and therefore save the cost of processing these

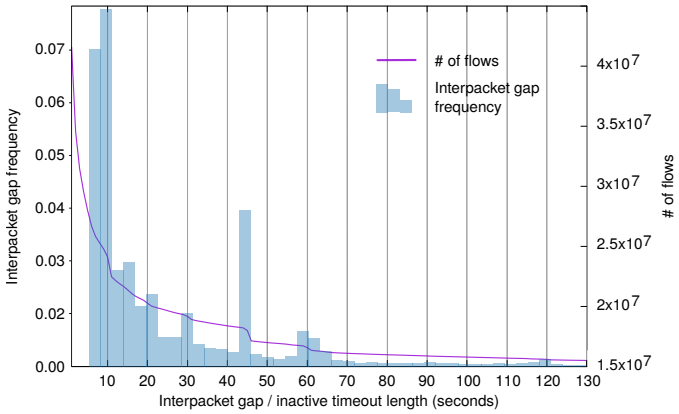


Fig. 1. Impact of interpacket gaps and inactive timeouts on number of flows. *(Dataset1, TCP)*

records on the collector and creating new ones during the flow measurement process.

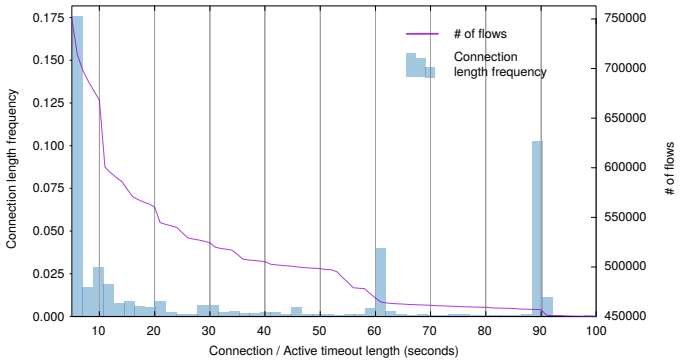
B. Impact of the Active Timeout

The active timeout splits long connections to flow records of maximum length of the value of the active timeout. Therefore, we can expect a correlation between lengths of connections, active timeout and the number of flow records. We captured this relation for TCP protocol connections of Dataset2 in Figure 2a. However, the correlation is not nearly as strong as that of interpacket gaps and inactive timeout. The reason is that the active timeout does not cause the flow records to be of exactly equal lengths. Instead, it terminates a flow record after the given time, and the flow record describing the rest of the connection starts at the next packet. Therefore, the sum of lengths of flow records split by an active timeout is less than that of the original connection. This means that the number of flows record when the active timeout is used depends not only on connection lengths but on interpacket gaps as well. Therefore, it is not as straightforward to find the relation between active timeout and number of flows.

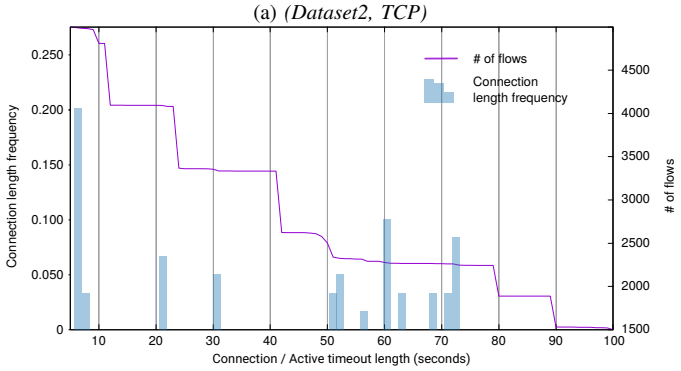
When the connection length is a multiple of the value of the active timeout, we can expect a change in the graph as well. This is the most likely reason for the large decrease in the number of flows for the active timeout of length 10 in Figure 2a.

If the packet in all connections were equally distributed in time (i.e. had the same interpacket gaps), the correlation between connection lengths, active timeout, and the number of flows would be stronger. The TCP protocol connections in Dataset2 have smaller interpacket gaps than the ICMP connections (see Table III). Figure 2b shows the same correlation for the ICMP protocol. Since longer interpacket gaps are more frequent for ICMP, the correlation is even weaker.

Based on these results, it is not feasible to determine the best active timeout based on the connection lengths found the observed network data. The active timeout configuration should rather be set based on external factors, such as the required timeliness of the data.

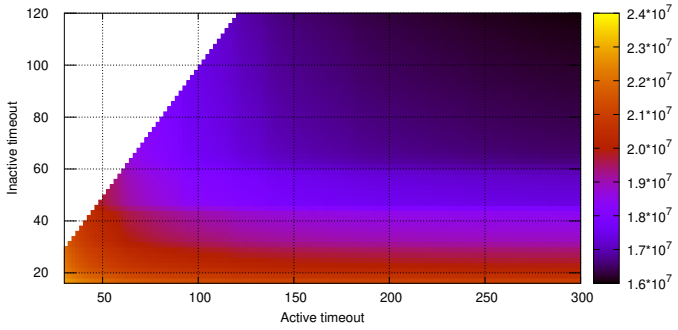


(a) *(Dataset2, TCP)*

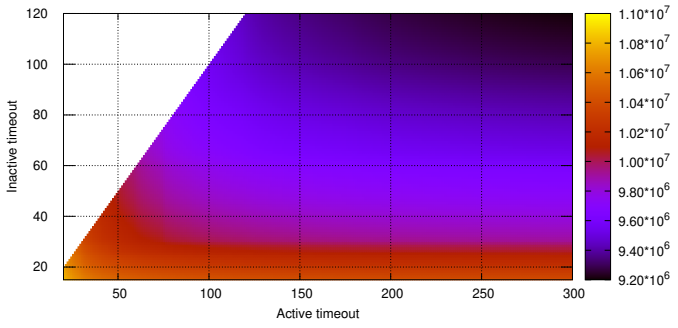


(b) *(Dataset2, ICMP)*

Fig. 2. Impact of connection lengths and active timeouts on number of flows.



(a) *(Dataset1, TCP)*



(b) *(Dataset1, UDP)*

Fig. 3. Heatmap of number of flows based on active and inactive timeout.

C. Impact of the Active and Inactive Timeout Combination

Active and inactive timeouts are used both at the same time in flow monitoring systems. We have plotted the relationship between timeouts and number of created flows using a heatmap where the colour represents the number of flow records. Figures 3a and 3b show the relations for the Dataset1 TCP and UDP protocols respectively. The range of inactive timeout was limited to $16 - 120$ s, the range of active timeout to $30 - 300$ s to highlight the most interesting parts. The horizontal lines in Figure 3a correspond to frequent interpacket gaps in Figure 1. As for the active timeout, we are able to discern changes at values 60, 90, 120, and 180. These values are given by common implementations of application protocols. However, for the active timeout larger than three minutes, there are are only gradual changes in the number of flows. Therefore, the active timeout configuration can be based on the preference of data analytics or capabilities of a flow collection software, rather than solely on the properties of the observed traffic.

Figure 3b shows that the behaviour of UDP protocol differs from the TCP. The heatmap exhibits more gradual transitions, which indicates a lack of clearly defined timeout values. Therefore, we expect that finding optimal flow expiration timeouts will differ for different transport protocols.

D. Impact of Timeouts on Slowloris Detection

The Slowloris attack, represented by the Dataset3 in this paper, is a DoS attack aimed to overwhelm the target HTTP server by sending and maintaining requests for the largest possible amount of time. This is achieved by sending the request header in small parts but never actually completing the request. We describe the connections generated by this attack and show how the detection of the attack is usually performed. Then, we will show that changing the flow expiration timeouts will break such a detection.

A successful attack begins by establishing a connection via the TCP three-way handshake. After that, the attacker sends the first part of the HTTP GET request header and continues to send a small part of the HTTP header every 100 seconds. This continues until the server responds with Bad Request HTTP error code, which takes more than 2470 seconds in our dataset.

When the server is successfully attacked, it first stops responding to HTTP requests. The TCP connections are still created, but the attacker does not get a response for the initial part of the HTTP GET request and terminates the connection after approximately 108 seconds. When the server is impaired by the attack even more heavily, it simply does not respond to TCP connection establishment attempts, and the attacker stops trying after sending three SYN packets in 3 seconds.

Most of the traffic in the Dataset3 are connections from one of the three described categories. Figure 4 shows numbers of flow records created for different active and inactive timeouts on the Dataset3. The breakpoint at active timeout 108 corresponds to a large number of failed HTTP connections. The same reason is for the difference at active timeouts 54, 36, and 27 as they are fractions of 108 (108 divided by 2, 3, and 4). The distinctive inactive timeouts correspond to application-specific timeouts after which the unresponsive sessions are closed. The inactive timeout of 100 corresponds to the interpacket gaps in the successful Slowloris attack.

The detection of the Slowloris attack can be based on

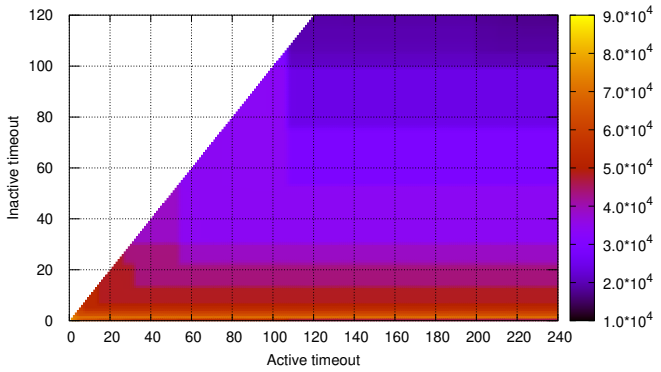


Fig. 4. A heatmap of timeouts and number of flow records for the Slowloris attack. *(Dataset3, TCP)*

flow monitoring data. It usually includes analysis of the attack behaviour and proposition of metrics that match this behaviour as close as possible to provide high detection rate and low false-positive rate. Machine learning algorithms are often utilised for the feature selection and for finding the optimal model. The authors of [11] have used exactly this approach. Some of the presented models use flow duration as a feature that helps to identify attacks. For example, one of the models requires that the connection is longer than 907 seconds. However, we have shown that the attacks can have more than 100-second interpacket gaps. Therefore, using flow records created with inactive timeout lower than 100 seconds would create flow records that do not match this model, and the detection would fail. Aggregating the flow records to create records closer to the original connections can mitigate this issue. However, this causes the detection to be postponed until after the attack is complete, which can take very long in the case of the Slowloris attack.

Another example of flow record utilisation for slow HTTP attack detection can be found in [12]. The authors use NetFlow features such as flow duration, number of packets, packet per second, and bytes per second. All these features heavily depend upon the flow expiration conditions. Although the authors describe the executed attacks in detail, the flow monitoring configuration is omitted. Based on our results, we can expect the outcome of the presented experiments to change significantly with any change to the flow monitoring expiration timeouts. Moreover, training the models using different timeouts than those that are used for subsequent evaluation could cause a decrease in the performance of the models.

VI. Discussion

The analysis of flow expiration timeouts shows that choosing the optimal timeout values is not a straightforward task. The timeout configuration is always a trade-off between flow completeness, timeliness, and resource utilisation. Although there is no simple guide for determining the flow expiration timeouts, it is useful to consider the following questions when choosing the timeouts:

Active timeout: What is the requirement on the timeliness of the data? Is any action taken in real-time based on the data? Does the flow collection tool require certain timeouts? E.g. the popular *NfSen* flow processing toolset [13] expects the active timeout to be 300 seconds.

Inactive timeout: How fast should the inactive connections be reported? Is the performance of the flow monitoring system (especially the amount of utilised memory) an issue? If it is, consider lowering the inactive timeout. Are there any application-specific timeouts that should be taken into consideration to avoid unnecessary dividing the flow records?

When determining the flow expiration timeouts, it might be useful to analyse the observed traffic. We have shown that different datasets exhibit different behaviour; therefore, using different timeouts would provide higher data quality. The presented tool [10] can be used to analyse a sample dataset and find timeouts for which the number of flows changes significantly. This knowledge can be used to discover application-specific behaviour and decide which expiration timeouts to use.

The analysis of the Slowloris attack and its detection method showed that the flow expiration timeouts influence the results of the methods based on time-related flow features. The lack of consideration for the configuration of the flow monitoring systems can be found throughout the research literature, e.g. [14], [15]. Moreover, not only attack detection is concerned. Any use of machine learning based on time-related flow features, such as traffic classification or identification is affected.

VII. Conclusions

In this paper, we have analysed the impact of flow expiration timeouts on the number of resulting flow records using different publicly available datasets. We have found that different types of traffic exhibit different behaviour with regard to expiration timeout settings. This behaviour differs between datasets as well as between different transport protocols. Therefore, it is advisable to set the timeouts according to the desired results based on the monitored traffic.

On an example of Slowloris attack, we have shown how the flow expiration timeout configuration affects the results of the attack detection method. This finding can be generalised for all algorithms using flow monitoring data and depending on time-based flow features. We have observed that researchers often neglect to disclose the flow monitoring configuration parameters of their experiments, which hinders reproducibility of their results. We believe that the findings presented in this paper will convince authors of such experiments to include the details about the flow monitoring configuration in their work. For this reason, we have proposed a series of questions that can help with selecting the correct flow expiration timeouts for a given scenario.

Acknowledgment

The publication of this paper and the follow-up research was supported by the ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/ 16_019/0000822).

REFERENCES

[1] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," RFC 7011 (INTERNET STANDARD), Internet Engineering Task Force, Sep. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc7011.txt

[2] G. Sadasivan, N. Brownlee, B. Claise, and J. Quittek, "Architecture for IP Flow Information Export," RFC 5470 (Informational), Internet Engineering Task Force, Mar. 2009, updated by RFC 6183. [Online]. Available: http://www.ietf.org/rfc/rfc5470.txt

[3] J. Quittek, S. Bryant, B. Claise, P. Aitken, and J. Meyer, "Information Model for IP Flow Information Export," RFC 5102 (Proposed Standard), Internet Engineering Task Force, Jan. 2008, obsoleted by RFC 7012, updated by RFC 6313. [Online]. Available: http://www.ietf.org/rfc/rfc5102.txt

[4] K. C. Claffy, H.-W. Braun, and G. C. Polyzos, "A Parameterizable Methodology for Internet Traffic Flow Profiling," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1481–1494, Oct. 1995.

[5] Í. Cunha, F. Silveira, R. Oliveira, R. Teixeira, and C. Diot, "Uncovering Artifacts of Flow Measurement Tools," in *Passive and Active Network Measurement: 10th International Conference, PAM 2009, Seoul, Korea, April 1-3, 2009. Proceedings*, S. B. Moon, R. Teixeira, and S. Uhlig, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 187–196.

[6] R. Hofstede, I. Drago, A. Sperotto, R. Sadre, and A. Pras, "Measurement Artifacts in NetFlow Data," in *Passive and Active Measurement*, M. Roughan and R. Chang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–10.

[7] J. M. Rodriguez, V. C. Español, P. B. Ros, R. Hoffmann, and K. Degner, "Empirical Analysis of Traffic to Establish a Profiled Flow Termination Timeout," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Jul. 2013, pp. 1156–1161.

[8] CAIDA, "The CAIDA UCSD Anonymized Internet Traces 2015 - 05/21," Mar. 2015. [Online]. Available: http://www.caida.org/data/passive/passive_2015_dataset.xml

[9] Communications Security Establishment (CSE) & the Canadian Institute for Cybersecurity (CIC), "A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)," Feb. 2018. [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018/

[10] P. Velan, "Flow Expiration Timeouts," Sep. 2019. [Online]. Available: https://github.com/CSIRT-MU/FlowExpirationTimeouts/tree/noms2020

[11] C. Kemp, C. Calvert, and T. Khoshgoftaar, "Utilizing Netflow Data to Detect Slow Read Attacks," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, July 2018, pp. 108–116.

[12] C. L. Calvert and T. M. Khoshgoftaar, "Impact of class distribution on the detection of slow HTTP DoS attacks using Big Data," *Journal of Big Data*, vol. 6, no. 1, p. 67, Jul 2019.

[13] P. Haag, "NfSen," Dec. 2011. [Online]. Available: http://nfsen.sourceforge.net/

[14] L. Canuto, L. Santos, L. Vieira, R. Gonçalves, and C. Rabadâo, "CoAP Flow Signatures for the Internet of Things," in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, June 2019, pp. 1–6.

[15] L. Vieira, L. Santos, R. Gonçalves, and C. Rabadão, "Identifying Attack Signatures for the Internet of Things: An IP Flow Based Approach," in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, June 2019, pp. 1–7.