Asynchronous Record Alignment of Network Flows for Incident Detection and Reconstruction

Virgilijus Krinickij and Linas Bukauskas
Institute of Computer Science, Vilnius University, Vilnius, Lithuania

virgilijus.krinickij@mif.vu.lt
linas.bukauskas@mif.vu.lt

Abstract: In today's interconnected digital landscape, the distribution of cyber threats presents a significant challenge to cyber security. Moreover, as of 2016, the amount of data in the world exceeds one zettabyte. Because of this, evidence-based network flow analytics is a critical component of modern network management and security. Problems such as anomalies in the network flow, cyber security incidents, alert generation, data pre-processing, network monitoring, network flow complexity, and data flow patterns become difficult to detect in massive network data flows. These specific problems can be addressed using Packet capture (PCAP). PCAP analysis is a standard network forensics process and investigation for assessing network behaviour and identifying anomalies. This work presents a method for analysing network flows for probable alignment of asynchronously recorded communications in heterogeneous networks. Using a proposed method for alignment, we can identify the relevant recordings aligned over two data streams for faster and more conclusive incident analysis. We use synthetic network incident scenarios for research experiments, detailing the generation of cyber event data and impact on cloud network traffic, followed by in-depth PCAP analysis. The automated cyber-attacks are simulated within a network infrastructure generating network flows in a PCAP format. Simulated cyber-attacks range from standard port scans, service scans, and specific scenarios like SQL injection, phishing, DoS or DDoS. We define analysis objectives and criteria for the in-depth PCAP analysis and alignment. The evidence gathered showcases valuable information about network data flow and its behaviour.

Keywords: Cybersecurity, Networks, Flow, Incident, Detection.

1. Introduction

Cyber warfare is an ever-evolving landscape that needs strong cyber security practices and strategies. Today, organisations and governments put huge efforts in time and money to secure their infrastructures from malicious actors. This became sort of a new, unprecedented arms race that you cannot physically touch (Clarke, R. A., & Knake, R. K., 2014). The definitive aspect of the actors getting ready is in network capabilities (Argyraki, K., & Cheriton, D., 2005). In one particular example, the more data can be produced for pushing through the network, the more possible harm it could generate (de Neira, A. B., Kantarci, B., & Nogueira, M, 2023). Because of this, Google has seen the biggest DDoS attack that was conducted against them last year. Huge amounts of data generated, gave a new view of how multi-vector attacks can overlap with additional possible false-positive or true-positive anomalies and other network problems. This means that trying to identify specific problems and analyse traffic that was accumulated in millions of requests or any other type of data flow through the network right now is only considered (Zahid et al, 2022, Dimolianis et al, 2019, Kalinin, M. O., & Krundyshev, V. M., 2021). For this, more detailed and feasible approaches are wanted by governments and organisations. One of the specific possible solutions is generated PCAP analysis. PCAP for forensics, troubleshooting network problems, or security-related questions are very helpful. They help many security professionals investigate particular cases of cyber events and more (Cappers et al, 2018).

Although there are a lot of PCAP formats (Veselý, V., 2012), they are mostly used for computer network communications. Of course, like every technology, PCAP comes with its own flaws. One of the flaws is the file size. Capturing even thirty seconds of network stream can generate a file that will be quite big in size. Most of the files come for live data capture and PCAP files are used for diligent data analysis.

One important aspect of PCAP analysis is packet alignment, which involves organising the packets based on specific parameters or attributes. Packet alignment helps to establish relationships between packets, align packets in a specific order for references and extract meaningful information from the captured data. By aligning packets based on relevant attributes, we can perform a more established and focused analysis.

Incident detection and reconstruction in PCAP analysis is a critical component of modern network management and security. In the context of Network security, network flow analytics helps to check network flow and troubleshoot for specific problems. The key motivational problems for this research include anomaly detection, forensics, compliance and reporting, policy enforcement, and real-time monitoring. Also, it is imperative to

understand that this method's results provide insights needed to make informed decisions, enhance security posture, optimise network performance, and ensure compliance with regulations and policies.

Any organisation or government would like to enhance their posture in cyber security by actively detecting and mitigating anomalous network behaviour that may indicate a security threat. The strategic steps for this scenario would be:

- Data Collection
- Data Pre-processing
- Data Analysis
- Anomaly Detection in Data
- Alert Generation

The goal is to identify unusual patterns and potentially malicious activities within the network. The generated results would help to increase the integrity of transmitted data and filter our unwanted patterns.

2. Related Work

The asynchronously captured data of a network flow in big quantities, hides the possible cyber-attacks or multi-vector attacks. This accumulates a deficiency that can be addressed with the help of PCAP analysis. On the other hand, synchronous alignment methods assume a uniformity and predictability in network traffic that is rarely encountered in real-world scenarios. Such methods are constrained by their requirement for simultaneous data availability, leading to potential delays or inaccuracies in incident detection when faced with asynchronous or out-of-order network flows.

One of these methods is the Euclidean Matching. The finite amount of points generated in a PCAP file may differ in count which must be equal based on the Euclidean matching problem. The other major issue is that the generated PCAP files are recorded asynchronously, where Euclidean matching should be symmetrical in many cases (Chew et al, 1997).

Another problem is the heterogeneous aspect of different network points used for network flow capture. For homogeneous alignment algorithms like Needleman-Wunsch from bioinformatics are used but this approach, however, faces challenges when dealing with heterogeneous packets (Beddoe, M. A. 2004). Also, the segmented-based alignment method introduced by (Esoul, O., & Walkinshaw, N, 2017) marks a significant advancement, employing segments from packets within specific protocols for alignment, but faces the same heterogeneity problem.

In research provided by (Diab et al, 2019) DTW (Dynamic Time Warping) algorithm is used for finding the best alignment in two time series with Euclidean distance involved for anomaly detection. Although we do not focus specifically on anomaly detection, the DTW method can be focused for asynchronicity in generated PCAP files per our synthetic scenarios. Also presented by (Diab et al, 2021) DTW can be used for TCP DoS/DDoS attack detection. While our scenarios presented later on will have this type of attack involved, we use different parameters from the generated PCAP files for our analysis. In addition, our scenarios vary greatly in time and specific attack details that are launched automatically from different network points. Also, based parameters are assigned to a specific case. The case pre-defines what we are looking for and want to look specifically in the generated PCAP files. For generated scenario incident detection, we address the provided case and parameters while using DTW algorithm.

3. Method

This section will present a network flow capture method.

In figure 1 we present data capturing using PCAP. In a controlled network environment, we use a master machine that manipulates two slave machines through an SSH connection. All the machines are virtual machines prepared in a bare metal hypervisor. The SSH connection is established asynchronously through a programmed code that is launched in the master machine. After the connection is made a program called *tshark* is launched for packet capture on the eth0 (Ethernet) interface in both slave machines. For packet capture we use *tshark*, a program used for packet capture and analysis (Merino, B., 2013). Additionally, while launching *tshark* we set a specific packet capture interval, which can be changed, and we specify the file, where packet capture will be stored.

Changing the capture packet interval ensures asynchrony in the packet capture. When the packet capturing process is initialised we launch a prepared attack from the threat actors machine to the target machine.

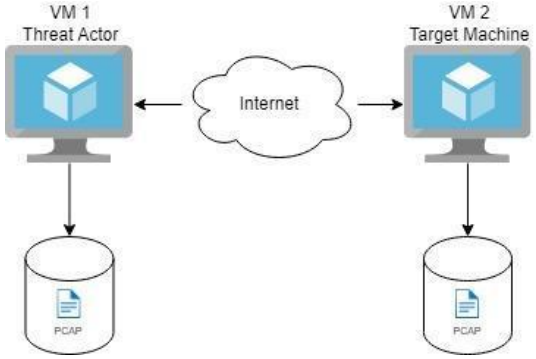

Figure 1: Data Gathering Using Virtual Machines

For the attacks we use synthetic network incident scenarios for research experiments. As mentioned before, we use different scenarios for data gathering. We generated 14 different synthetic scenarios for experiments. The primary scenarios are through a network port scan of TCP SYN packet DoS attack with different detail variations. Example scenarios may include:

Scenario A, Taking the full spectrum of ports from 1 to 65535.

Scenario B, Taking ports from 1-10000.

Scenario C: Taking the most common ports used.

Scenario D: Taking ports from 1-10000 where a proxy machine is used to attack the target machine.

Other scenarios.

Scenario A and B, is a standard scan with no additional prerequisites, while scenario C is used only for the most relevant ports today. Scenario D adds payloads that could be executed in the target machine later on. For the attacks, we use the *Nmap* (Network Mapper) tool. *Nmap* helps to generate the needed number of requests for the attack. While sending a massive number of TCP SYN packets without completion of the TCP handshake, the TCP SYN attack is regarded as a DoS attack. This helps in scenario D where we denote that this is a secondary type of a technique used on a specific machine. Other scenarios are different attacks like an SQL injection, and so on.

During the execution time, both machines and their listeners are active and data flow is captured on them. When the attack is over, the programmed code in the master machine tells both, the threat actor machine and the target machine to wait for extra predefined time, to stop the flow capture and save the captured flow in PCAP files in respective machines. Then the files are transferred back to the master machine for PCAP analysis.

All of the actions described are done automatically at heterogeneous network points with different visibility. Meaning that the programmed code is launched as a script written in bash where we describe the logic of automatically launching the SSH connections, the attacks and packet capturing.

4. Algorithm

This section will present an alignment algorithm for PCAP file analysis.

The raw data that comes in from a network stream to the PCAP, is very hard to read. For this, data pre-processing is the essential step in PCAP analysis (Alothman, B., 2019). Depending on the goals set for data pre-processing, the method may vary. In figure 2 we use these PCAP pre-processing steps.

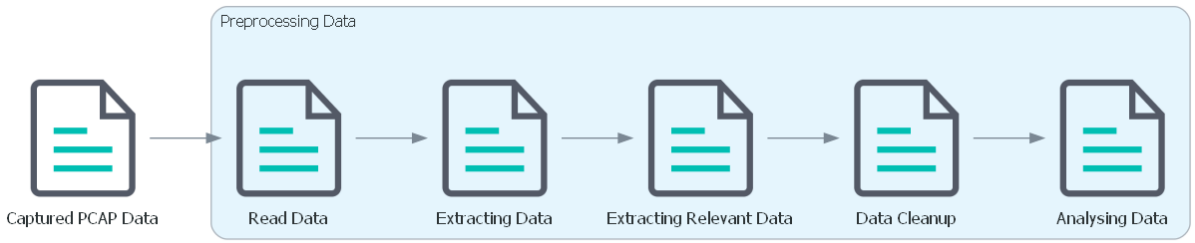

Figure 2: PCAP Pre-processing Method

The pre-processing steps are needed to ensure that at least some amount of noise will be deducted.

In figure 3 the two generated PCAP files are presented as a scenario that will be given in a later step to DTW (Dynamic Time Warping Algorithm) (Senin, P., 2008). The DTW here is used for aligning both network flows in PCAP files based on specific cases and additional parameters. The DTW works with sequences:

$$x=\{x1, x2, x3, \ldots, xi, \ldots, xn\}$$

$$y=\{y1, y2, y3, \ldots, yi, \ldots, yn\}$$

These sequences are then given to the DTW for creating a warping path and distance. The warping path is the path that is calculated to find similarities and the distance is the difference between the two sequences presented to DTW. The warping path is calculated by creating a matrix, where each element represents the distance between two sequences. Then a path is searched within the matrix that minimises the total cumulative distance. To do this, the algorithm compares each point of one sequence to every point of another. While comparing, additional constraints can be added to find the best match to form a path in the matrix.

DTW is used as a primary algorithm versus the classical Euclidean matching (Chew et al, 1997).

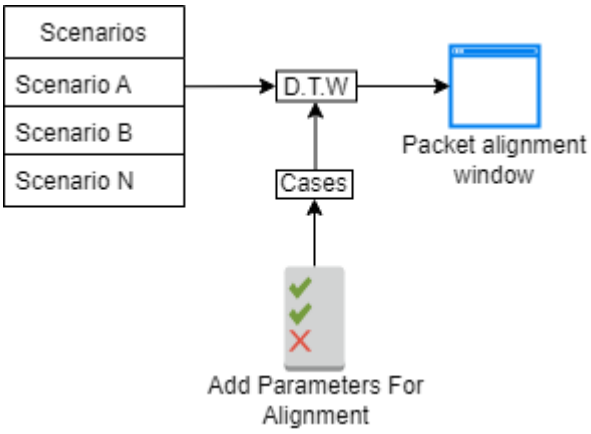

Figure 3: PCAP Alignment Algorithm

While adding scenarios to the DTW, we assign a case that we wish to explore in the scenarios for any A and B sources there exists content(A) ≈ content(B) ∧ content(B) ≈ content(A), here A, B are records in time. This implies that we are looking for single coincidences in the content A and B when A and B are records in time.

While processing gathered data, we add parameters for alignment. Based on the scenarios, we choose the TCP protocols flags and unit ports in content as parameters per the provided cases, that will be the key components submitted. Finally, after examining the PCAP file through an iterative process, we create a packet alignment window in a graphical form, to show the results.

The flags taken are SYN-ACK and RST-ACK. Then we check if the source port corresponds to the destination port from both flags respectively. The ports are taken as a unique unit value from both flag arrays, and are matched between the flags. The pseudocode for these functions is presented in figure 4 and figure 5:

Algorithm 1 Extract Packets With Needed Flags

```
Require: pcap_file
  packets ← readFile                          ▷ Read packets from file
  ack_rst_packets ← []                        ▷ Initialize empty list
  for each packet in packets do
      if packet has a TCP layer then
          if packet TCP flags are AR or SA then
              Append packet to ack_rst_packets
          end if
      end if
  end for
      Return ack_rst_packets
```

Figure 4: Pseudocode for Extracting Packets Based on Flags.

Algorithm 2 Match Ports

```
Require: ack_rst_packets
  packets1 ← []                               ▷ Initialize empty list
  packets2 ← []                               ▷ Initialize empty list
  for each packet in packets do
      if packet has a TCP layer then
          if packet TCP flags are AR or SA then
              Append packet to ack_rst_packets
          end if
      end if
  end for
      Return ack_rst_packets
```

Figure 5: Pseudocode for Matching the Ports From the Given Packets.

The generated PCAP files varied in size from 200 kilobytes to 14 megabytes. As previously mentioned, an increase in file size directly correlates with a corresponding increase in the volume of packets encapsulated within the PCAP file. Furthermore, the computational demands associated with the packet processing task were demanding. Standard computing power units were used for the packet processing task described. The time to complete the task varied from 10 seconds to 2-3 hours. One specific, not defined scenario in this research that accumulated a 360 megabyte ~759 000 packet PCAP file took approximately 6 hours to process.

5. Results

This section will present the results of the algorithm.

Frequent attack pattern in alignment was spotted at the start of the attack, with subsequent alignments appearing consistent throughout. Assuming that the most frequently used ports would be only addressed per the most frequent port scan scenario mentioned before, indicates that this pattern can also be addressed in other scenarios not depending on details provided (figure 6 and 7). For graphical representation of the results we use python library (Meert et al, 2020).

The alignment that appears consistent throughout suggests that the attack pattern changed because the needed results of the attack were achieved or the firewall noticed the attack pattern and mitigated the attack leaving the static data flow until the end without any results. Even though the same ports were triggered multiple times after the initial alignment, no considerable changes were seen. As mentioned before the bare metal hypervisor used for the experiments is a part of Vilnius University high performance computing infrastructure and no additional firewall configuration was made for experiments.

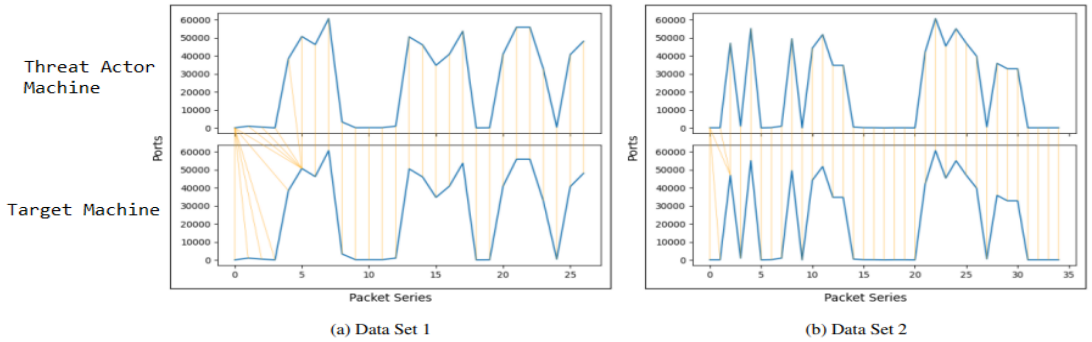

Figure 6: Asynchronous PCAP Alignment from SA-RA Packets and Source Destination Ports 1

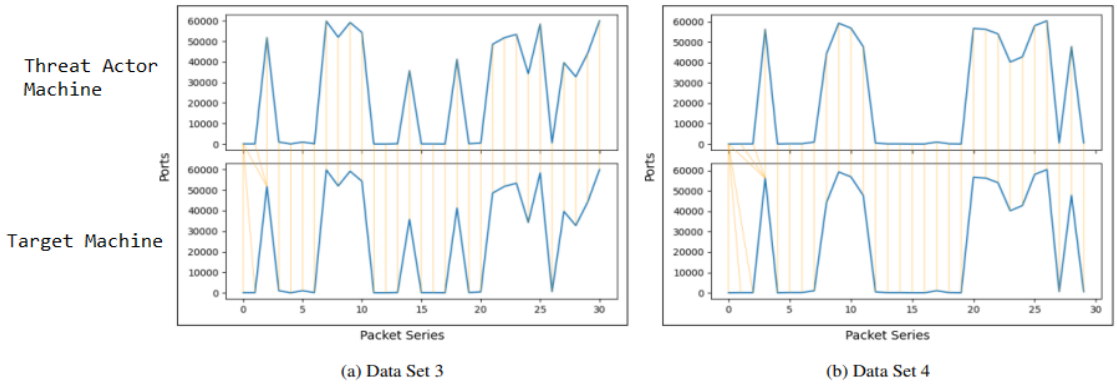

Figure 7: Asynchronous PCAP Alignment from SA-RA Packets and Source Destination Ports 2

Furthermore, this can be seen in the small amount of packet series accumulated between the two PCAP files. Although the SA-RA packets were gathered at the start of the attack which generated the alignment, only static callback was seen afterwards. Also, the small packet amount of packet series could indicate that other ports per the 60000 ports scanned were closed or not reachable based on some additional firewall rules or other network configuration.

Previously accumulated results noticeably change in one specific scenario. In these results the amount of accumulated SA-RA packets exceeds the previously mentioned amount (figure 8).

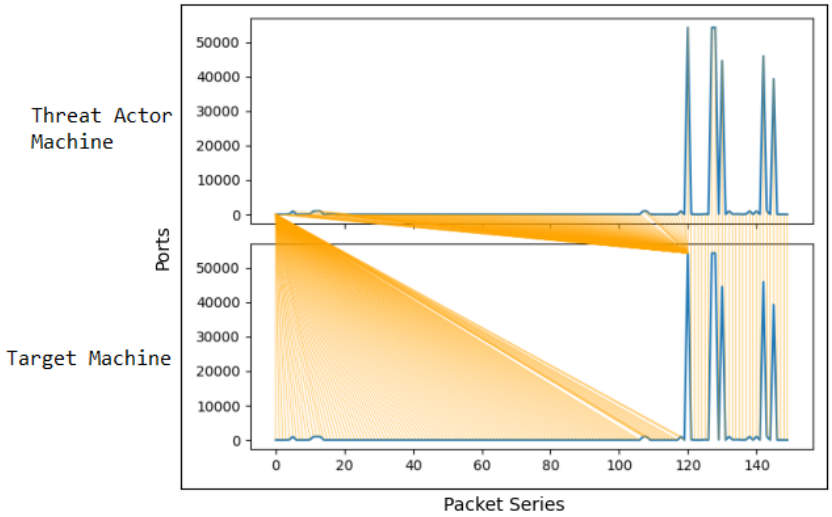

Figure 8: Asynchronous PCAP Alignment from SA-RA Packets and Source Destination Ports 3

We see two asynchronous network flows. The difference in amount is noticed by the yellow lines that represent SA-RA trigger count per ports which aligns the two sequences in time. The blue lines show the port number that was used to call in the attack. Possibly due to heterogeneous network points used or a very specific payload that bypassed default defence mechanisms, the accumulation of the packet series between two files became noticeably bigger. The number of ports affected was not in a large number, but they were seemingly attacked in a disproportionate amount.

6. Discussion

The realised pattern in the results suggest that the attack from target machine to victim machine generates an attack pattern based on scenarios provided.

Also, based on the parameters selected we see that the alignment between two-time frames while using DTW was achieved in the detection and reconstruction of the attacks in asynchronous records. The definitive distinction comes in the computational power and buffer problem in DTW. DTW can be less effective for PCAP alignment because it needs to access the full data stream of packets, which must be buffered before the procession. The exhausting nature of the algorithm causes the deficiency in computational power. In time this buffering can delay the potential of DTW to do immediate data analysis or alignment.

If these problems are addressed, the possible alert generation could be sufficient in detecting known vulnerabilities. For sophisticated vulnerabilities additional steps would be needed. These steps may include artificial intelligence or machine learning components implemented.

Based on the previous research, heterogeneous point selection of the network was implied, less than possible. Furthermore, it is worth noting, that while these patterns are suggestive of malicious activity, they should be interpreted with caution, considering the potential for false positives in complex network environments.

It is possible to have different configurations in different heterogeneous network points. These configurations can come out in the same network's subnets, DMZ and other solutions.

7. Conclusions and Future Work

The research conducted in this study provides a fundamental understanding of asynchronous evidence-based assessment, the analysis of network traffic for cybersecurity.

For a broader spectrum of problems, we assume the potentially heavy cyber-attacks that are masqueraded by large volumes of data flow. This comes down to pre-processing and processing relevant data. As we experienced a struggle with small sized data structures, big PCAP analysis would require much more computational power. The DTW algorithm does not provide the sufficient results needed. As a result, future studies should focus on the scalability and efficiency of security models in environments with high volumes of network traffic.

Furthermore, future research will possibly focus on developing more sophisticated models for asynchronous evidence-based assessment. This includes automatization in incident response systems based on previously mentioned artificial intelligence and machine learning involved. Also, a deeper dive in network configuration that could enhance the understanding of network flow in correlation with cyber-attacks can be discussed. This is needed if we want to understand the different visibility of the networks that data flows from.

References

Alothman, B. (2019, June). Raw network traffic data preprocessing and preparation for automatic analysis. In *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)* (pp. 1-5). IEEE. http://dx.doi.org/10.1109/CyberSecPODS.2019.8885333

Argyraki, K., & Cheriton, D. (2005). Network capabilities: The good, the bad and the ugly. ACM HotNets-IV, 139, 140.

Beddoe, M. A. (2004). Network protocol analysis using bioinformatics algorithms.

Cappers, B. C., Meessen, P. N., Etalle, S., & Van Wijk, J. J. (2018, October). Eventpad: Rapid malware analysis and reverse engineering using visual analytics. In *2018 IEEE Symposium on Visualization for Cyber Security (VizSec)* (pp. 1-8). IEEE. http://dx.doi.org/10.1109/VIZSEC.2018.8709230

Chew, L. P., Goodrich, M. T., Huttenlocher, D. P., Kedem, K., Kleinberg, J. M., & Kravets, D. (1997). Geometric pattern matching under Euclidean motion. *Computational Geometry*, *7*(1-2), 113-124. https://doi.org/10.1016/0925-7721(95)00047-x

Clarke, R. A., & Knake, R. K. (2014). *Cyber war*. Old Saybrook: Tantor Media, Incorporated.

de Neira, A. B., Kantarci, B., & Nogueira, M. (2023). Distributed denial of service attack prediction: Challenges, open issues and opportunities. *Computer Networks*, *222*, 109553. https://doi.org/10.1016/j.comnet.2022.109553

Dimolianis, M., Pavlidis, A., Kalogeras, D., & Maglaris, V. (2019, April). Mitigation of multi-vector network attacks via orchestration of distributed rule placement. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)* (pp. 162-170). IEEE.

Diab, D. M., AsSadhan, B., Binsalleeh, H., Lambotharan, S., Kyriakopoulos, K. G., & Ghafir, I. (2019, August). Anomaly detection using dynamic time warping. In *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)* (pp. 193-198). IEEE. https://ieeexplore.ieee.org/document/8919604/

Diab, D. M., AsSadhan, B., Binsalleeh, H., Lambotharan, S., Kyriakopoulos, K. G., & Ghafir, I. (2021). Denial of service detection using dynamic time warping. International Journal of Network Management, 31(6), e2159.

Esoul, O., & Walkinshaw, N. (2017, July). Using segment-based alignment to extract packet structures from network traces. In *2017 IEEE International Conference on Software Quality, Reliability and Security (QRS)* (pp. 398-409). IEEE. https://doi.org/10.1109/QRS.2017.49

Kalinin, M. O., & Krundyshev, V. M. (2021). Analysis of a huge amount of network traffic based on quantum machine learning. *Automatic Control and Computer Sciences*, *55*(8), 1165-1174. https://doi.org/10.3103/S014641162108040X

Meert, W., Hendrickx, K., Van Craenendonck, T., Robberechts, P., Blockeel, H., & Davis, J. D. Zenodo. 2020. https://zenodo.org/badge/latestdoi/80764246

Merino, B. (2013). *Instant traffic analysis with Tshark how-to*. Packt Publishing Ltd.

Senin, P. (2008). Dynamic time warping algorithm review. Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA, 855(1-23), 40.

Veselý, V. (2012). Extended comparison study on merging PCAP files.

Zahid, F., Funchal, G., Melo, V., Kuo, M. M., Leitao, P., & Sinha, R. (2022, July). DDoS Attacks on Smart Manufacturing Systems: A Cross-Domain Taxonomy and Attack Vectors. In 2022 IEEE 20th International Conference on Industrial Informatics (INDIN) (pp. 214-219). IEEE. http://dx.doi.org/10.1109/INDIN51773.2022.9976172