Network Traffic Anomaly Detection and Analysis – from Research to the Implementation

Slavko Gajin[1,*]

[1]School of Electrical Engineering, University of Belgrade, Bul. kralja Aleksandra 73, Belgrade, Serbia

Abstract

With a constantly increasing amount of encrypted network traffic and a new type of attack ("zero-day"), network traffic anomaly detection shows significant benefits over traditionally used signature-based packet inspection methods for cybersecurity attack detection. Using NetFlow or similar protocols is an attractive approach to providing accounting information about network communications due to its simplicity and applicability in a real-life network environment. Even though the basic set of information in flow data is not sufficient for efficient machine learning techniques, they are quite suitable for the application of entropy-based anomaly detection techniques. In this paper, we present comprehensive work in research, development and implementation of network traffic anomaly detection solutions based on the entropy of flow data. Starting from the well-known entropy-based approach, we reveal the results of our methodic work in solving the main challenges in designing an efficient anomaly detection solution empowered with the original classification method. Since the proof of concept was achieved in the laboratory environment using offline datasets, the solution has been implemented relying on the existing NetFlow Analyzer software product NetVizura. Even at the minimum viable product stage, the application confirms high performances and great applicability of the anomaly detection and classification method in real-life network environments.

Keywords

Anomaly detection, anomaly classification, entropy, network behaviour analysis

1. Introduction

The modern enterprise network environments face the necessity to respond adequately to the increased multi-heterogeneity needs, reflected through a range of different user devices, a variety of existing applications, services, and various data in different formats and throughputs, that need solid storage and real-time processing available. The rise of the global need for bringing office functionality to remote working personnel has additionally contributed to the need for the development and implementation of technologically mainstream-based network infrastructures, but also causing the appearance of new categories of cybersecurity threats. These challenging circumstances lead to the need for the adoption of a range of novel network security approaches, basing its strengths on the adoption of the „zero trust" security foundations as the main principle ("never trust, always verify").

The current cyber threat landscape indicates several interrelated activities starting from the intruder attempts to infiltrate, scanning, and collect data and infrastructure vulnerability information, intending to proceed with more severe attack routines. It relates to further data/infrastructure compromise and negative repercussions to operational abilities, productivity, privacy abuse, serious financial threats (ransomware), service provisioning issues, etc.

The European Union Agency for Cybersecurity, ENISA annually publishes reports on the cybersecurity landscape situation, and for 2022 the prime identified intrusions correspond to ransomware, social engineering threats, malware, intrusions targeting data and user privacy, attacks to the data and service availability (Denial of Service and Internet threats), disinformation attacks, encompassing the misinformation and the supply-chain attacks [1]. On the other side, the Nexus Guard 1HY 2022 report indicates a 75.6% of increase in the average attack size, with the maximum size values reaching more than 230 Gbps. There is a notable increase in UDP-based attacks (77.53%), application attacks (330%), and amplification attacks (106.65%) [2]. The predominant attack vector types correspond to the UDP group of activities, which contributed to almost 40% of all the detected attacks, while there was a large portion of the activity related to the HTTPS Flood intruders, with 16% of overall malicious activity. It is also noticed, that TCP ACK attacks had a relatively high part in these activities, with 6.5% of performed attacks. The rest corresponds to the TCP SYN, IP Fragmentation, UDP, DNS, L2TP, and SNMP amplification attacks. If analyzing attacks by category, the volumetric Direct Flood attacks contributed to 68% of the total attacks, while there was a noticeable increase of 330% of the application attacks activity, forming 17.5% of all the 2022 cyberattack landscape. The volumetric amplification attacks were also present, while in lower amounts than the other two categories.

BISEC'22: 13th International Conference on Business Information Security, December 03, 2022, Belgrade, Serbia

[*] Corresponding author.

slavko.gajin@rcub.bg.ac.rs (S. Gajin)

As reported by Truesec, the total number of cyber attacks increased by 160%, with the highest percentage for ransom (34%), access harvesting (32%), resource hijacking (12%), business email compromise (8%), data theft (7%). 40% of the attacks in the analyzed period originated from publicly accessible and vulnerable systems, with a noticeable rise of zero-day exploits and ransomware attacks [3].

This dynamic emergence of new-generation attacks, and the newly identified vulnerability targets, made perfect conditions for the increase in zero-day attacks, and different forms of amplification and application attacks, with persistent use of cryptography for the attack traffic generation. As a response, a large number of research groups and individuals are developing advanced and original solutions for accurate, efficient, and real-time intrusion detection. As there is no exclusive, all-encompassing solution for every incident case, network environment, or user need, the goal is to find specific prevention and detection methods that would fulfil the general range of modern network security requirements.

In anomaly detection, a considerable part of the research community is now directed toward the analysis of traffic behaviour and its characteristics, both structural and informative, searching for the correct identification of the anomalous from the normal instances. The traditional solutions rely on the use of firewalls configured for protection against well-known threats, which are usually combined with signature-based security models. However, new conditions and circumstances require additional measures which could bring intelligence, stronger analytics, and the highest possible response while processing massive volumes of relevant data. These techniques mostly rely on entropy-based and machine-learning algorithms.

In this paper, we present the work resulted from the project “Network Traffic Anomaly Detection system based on NetFlow data analysis – TRADE”, under the European EUREKA programme. The main project goal was to develop a competitive market-ready cybersecurity threat detection solution. To achieve this goal, initial requirements were defined for the conducted research and the associated software development as follows:

- Rely on unlabelled input data which should be easily collected in real-time
- A high data rate (tens of thousands of data items per second)
- High and reliable detection performances
- Raw data persistence
- Root cause analysis and deep drill-down data inspection
- Configurable, flexible and scalable solution.

The rest of the paper is organized as follows: the second section highlights the most relevant scientific work in this field. The third section presents the main solution, while section four discusses other important details and solutions in the proposed methodology, demonstrating the experimental results. Finally, the paper is concluded by summarizing the main achievements and results, and discussing further development.

2. Related work

Being one of the most significant research fields, cybersecurity and privacy provisioning have raised high standards for the accepted and applied techniques and approaches. Available literature provides access to significant research work focused on anomaly and cybersecurity attack detection, the so-called Intrusion Detection Systems (IDS). These papers are mostly published in the area of machine learning algorithms and are oriented towards the techniques for extracting the network traffic pattern’s behaviour.

The most common solutions are usually based on specific custom-made algorithms and procedures for inspecting network traffic and further detection and identification of network anomalies [4]. This category of IDSs relates to the analysis and proper results generation considering the specificities of the network traffic structure, patterns, and routines.

The very attractive research approach in designing and implementing real-time anomaly detection solutions is founded on the processing of flow-based data instances. When compared to the packet-inspection algorithms, this is in some aspects assumed as less accurate, but the experience indicates that these techniques, when combined with some additional methods can more easily and efficiently deliver accurate results.

In IDS, the entropy-based approach is easily implemented by relying on flow-based data and it has found a large portion of usability in situations when anomalous traffic appears with high intensity and volumes. In that case, it appears with weighty spikes in data distribution and does not leave these intrusions hidden [5, 6]. Some distinguished studies enforce the conception of the dominance of the parametrized entropies over the standard Shannon [7].

Still, our previous research results found that the published results in a high portion depend on the used detection techniques, data characteristics, and the choice of the analyzed features, thus we claim these conclusions groundless, as there is no possibility of their generalization [8, 9]. Nevertheless, the entropy-based techniques are powerfully useful in traffic stability analysis, challenging outlier detection, as well as in DDoS security provisioning [10, 11, 12].

Despite some statements that flow-based analysis relies on insufficiently detailed information, asserting the fact that these are only the basic information of network communications collected and exported by routers, the flow-based detection approach has gained strong enough approval among the research communities [13, 14].

The main issue with the application of entropy-based technics is some intrinsic deficiencies, being the most stumbling the fact that these techniques are eligible only when network traffic structure significantly changes through the attack [15, 16]. Another study analyzes the use of the real-time traffic behaviour profiling solution developed for the needs of anomalies and attack detection [17]. With a goal to detect particular behaviour patterns, the study is based on the analysis of traffic data, previously preprocessed with a set of entropy-based and data-mining techniques. This research relies on traffic aggregation and evaluates two additional attributes, the packet size and packet rate attributes.

The goal of our research was to propose a solution that makes the most of entropy-based techniques advantages and flow data collected in real-time while providing a high level of detection accuracy and efficiency, demonstrated through practical implementation.

3. Proposed solution

3.1. Flow-based approach

The access to network traffic provides a reach set of information for the analysis and anomaly detection, which includes both packet header and payload. However, this approach is not scalable and flexible enough, since it requires direct access to network resources. Also, it generates a huge amount of data, which requires high processing power for real-time analysis.

A more flexible approach is achieved by the Cisco NetFlow protocol, where routers keep track of each network communication, counting the transferred bytes and packets, and exporting this information to an external server. A large number of the collected data is treated as real-time logs about the communication activities, identified by the source and destination IP and port numbers, as well as the protocol used. Despite a lack of packet payload, these logs provide an excellent source of data for the analysis of network communication. The success of this approach was confirmed by a great acceptance by users and adoption on the market. Many other vendors proposed similar protocols, such as Jflow by Juniper Networks, Cflow by Alcatel-Lucent, NetStream by Huawei, Rflow by Ericsson, while IEEE has standardized IPFIX protocol [18].

So-called NetFlow Analyzer software collects the data, processes it and provides statistics about the traffic structure at the interfaces level, router level or in the whole network. Statistics are produced in bits per second, packets per second and flows per second metrics, given in a break-down structure of top contributors, such as the most intensive hosts, protocols, services etc. Additionally, NetFlow Analyzer keeps the logs in the internal database allowing manual drill-down analysis and data forensics. Therefore, network traffic anomalies can be efficiently investigated, but still hardly detected, with an exception of very intensive volumetric anomalies, such as DDoS attacks.

3.2. Entropy-based approach

To detect anomalies in network traffic the collected flow data must be first processed and transformed into a simpler form and metrics that provide a possibility to profile normal behaviour. Then, significant changes to normal behaviour are treated as an anomaly. Two main approaches to completing this task relate to machine learning and entropy calculation techniques.

Supervised machine learning is a very popular approach in the research literature, but the fact that it relies on labelled data gives very limited options for practical implementation in a real-time environment. Unsupervised machine learning for anomaly detection, mostly based on clustering methods, is more feasible for practical usage, but still very complex and demanding in data processing. For these reasons, the entropy-based approach attracts the interest of both the research community and industry. In the context of anomaly detection techniques entropy is a single value that can be interpreted as the measure of the evenness of a data distribution. A significant change in a data distribution causes a change in the entropy value, which is considered anomalous behaviour of network traffic and the indication of security threats.

Data distribution is obtained through the aggregation process using a selected attribute as an aggregation key and counting or summarizing other attributes from raw flow data. For instance, summarizing the total bytes or packets received by each destination IP address during a short period, the so-called epoch, will produce a corresponding data distribution. Typically, some addresses are more active than others, resulting in uneven data distribution and certain entropy values. In case of a DDoS attack, a targeting destination host receives a huge amount of traffic which cause a high peak in the data distribution and a significant change of the entropy value.

The flow data is identified by a flow tuple, which consists of the source and destination IP addresses, protocol type and source and destination port numbers. These identification attributes are good candidates for the aggregation keys and we will shortly label them as S, D, P, s and d respectively.

Since network communications between two pears mostly conduct in both directions, it is useful to pair two corresponding originally unidirectional flow data into a single bidirectional flow record [5]. The source address and port number are associated with the initiator of the conversation, while the destinate address and port number relate to the responding side. Consequently, the byte and packet counts relate to the sending side, either source or destination. These volumetrics attributes are labelled with sB, sB, sP and dP respectively. In the above mentioned example, the total bytes sent to a destination host is labelled as $sB[D]$. This metric, the so-called feature, relates to the data distribution and the corresponding entropy value for a certain epoch. During a time, epoch by epoch, the process generates a time data series of entropy values for each feature.

The Shannon entropy [19] is commonly used, while some other authors propose the usage of Tsallis [20] and Rényi [21] parametrized entropy types. For a given feature data distribution with the total number of N elements, where mi relates to the value of the element x_i and $p(x_i)$ is the empirical probability, the Shannon entropy is defined by the following equation:

$$H_S\left(X\right)=\sum\nolimits_{i=1}^{N}p\left(x_i\right)\log_b\frac{1}{p\left(x_i\right)} \tag{1}$$

where the empirical probability is given by:

$$p\left(x_i\right)=\frac{m_i}{M},\, M=\sum\nolimits_{i=1}^{N}m_i. \tag{2}$$

For an ideally even distribution with all equal elements, Shannon entropy reaches the maximum value of $\log_b N$, while more unevenness leads to lower, but always positive values. To always get the values between 0 and 1 it is useful to use the scaling factor of $1/\log_b N$.

The next challenge is to detect the changes in the entropy time-series values for any of the used features. To do so, the trend of the time series entropy data must be predicted based on the recent values, which is also known as the baseline values. Then, the acceptable variation in these values is calculated and set around the baseline and used as thresholds. A simple approach is the windowing mechanism setting the lower and upper thresholds by calculating the minimum and maximum values for the last N epochs. To make this accepted margin more tolerant, these thresholds are further increased by some scaling factor, usually by the value of 3.

Even though this approach dynamically adjusts the thresholds to recent variations of the observed values, a more advanced and flexible approach is based on the Exponential Moving Average (EMA) technique for short trend prediction [22], which provides better fine tuning options to adapt more accurately.

With the EMA technique, the baselined value in epoch n, labelled as $\hat{H}_n$, is predicted recursively, considering both entropy and baselined values in the previous epoch:

$$\hat{H}_n=(1-\alpha_h)\hat{H}_{n-1}+\alpha_h H_{n-1}. \tag{3}$$

The coefficient α_h, in the range between 0 and 1, is a weighting factor, to adjust the influences of the input values and make the baseline value more or less smooth.

The next step is to predict and baseline the standard deviation (S) of the entropy values, also using the EMA approach:

$$\hat{S}_n=(1-\alpha_S)\hat{S}_{n-1}+\alpha_S S_{n-1}. \tag{4}$$

And finally, the lower and upper thresholds are set relatively from the baselined entropy value $\hat{H}_n$, using a multiplication factor k_t, the so-called threshold factor, that makes the range wider:

$$\underline{T}_n=\hat{H}_n-k_t\hat{S}_n \tag{5}$$

$$\bar{T}_n=\hat{H}_n+k_t\hat{S}_n \tag{6}$$

The thresholds define a margin for acceptable variations:

$$T_n=\left[\underline{T}_n,\bar{T}_n\right] \tag{7}$$

The entropy values that fall into the margin T are considered regular, while the entropy value out of the margin triggers an alarm as an indication of an anomaly:

$$A_n=\begin{cases} True, & if\, H_n\in T_n\\ False, & if\, H_n\notin T_n\end{cases} \tag{8}$$

4. Other challenges and the proposed solutions

The main benefit of the entropy-based approaches lies in the fact that a complex data structure can be transformed into a different domain of time series data for the observed features that can be easier analyzed to detect unusual behaviour.

To be efficient for practical usage in real time environment anomaly detection solution requires proper feature selection (the aggregation keys and the calculating attributes) and solving other challenges related to the specific domain of network behaviour analysis based on collected flow data. In this section, we will demonstrate and discuss these challenges and propose proper solutions to them.

4.1. Baseline correction during the anomaly

The baseline calculation, either using EMA or sliding window techniques, takes into account previous values. In the case of an anomaly, detected by a drop in the entropy values, the unusually low entropy values gradually lead to lowering the baselined values too and widening the threshold margin. As a result, an anomaly could be eventually treated as normal behaviour. Consequently,

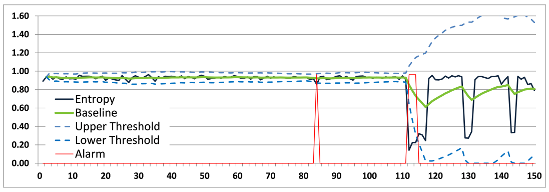

Figure 1: Baseline change during an anomaly (dataset CIC-IDS2017, trace Friday-PortScan, feature d[S]).

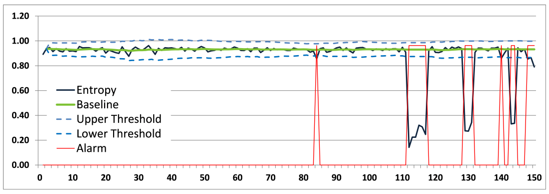

Figure 2: Baseline immutability during an anomaly (dataset CIC-IDS2017, trace Friday-PortScan, feature d[S]).

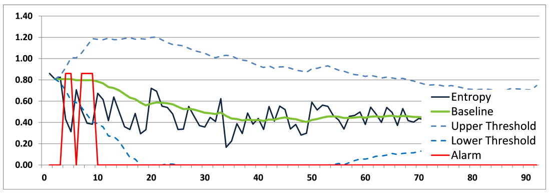

Figure 3: Large variations of the volumetric features (dataset CIC-IDS2017, trace Friday-PortScan, feature sP[S]).

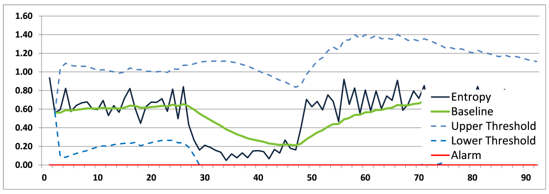

Figure 4: A wide threshold margin with a large initial standard deviation (dataset CIC-IDS2017, trace Friday-DDOS, feature s[D]).

when the anomaly stops, regular traffic could be treated as anomalous. It will take time, sometimes tens of epochs, for baseline and threshold values to get back into a normal range. This situation is demonstrated in Figure 1.

To get immune to the irregular entropy values during the long-lasting anomalies, we must ensure that the baseline of both entropy values, as well as the standard deviation values, are not affected when an alarm is raised, keeping the previous value unchanged, using the modified equations:

$$\hat{H}_n = \begin{cases} (1-\alpha_b)\,\hat{H}_{n-1} + \alpha_b H_{n-1}, & H_{n-1} \in T_n \\ \hat{H}_{n-1} & H_{n-1} \notin T_n \end{cases} \tag{9}$$

and

$$\hat{S}_n = \begin{cases} (1-\alpha_s)\,\hat{S}_{n-1} + \alpha_s S_{n-1}, & H_{n-1} \in T_n \\ \hat{S}_{n-1} & H_{n-1} \notin T_n \end{cases} \tag{10}$$

Figure 2 confirms that the proposed correction provides a stable and reliable prediction of normal behaviour.

4.2. Feature selection

Many authors in the scientific literature for DDoS attack detection propose entropy-based approaches that rely on the volumetrics attributes only, such as total bytes and packets number [23, 24, 25, 26]. However, our research has shown very limited usability of those approaches, mostly because regular traffic in today's network usage often assumes large data transfer, such as backup, data download or torrent communications. These occasional traffic loads cause a large variation in the corresponding entropy values and consequently a large threshold margin, which is shown in Figure 3. In some cases, the lower threshold can drop below the minimum value of zero, which makes it useless even for the detection of a high DDoS attack. Therefore, this kind of attack is easier to detect by inspecting top contributors in the data distribution, which is already achieved by NetFlow Analyzer software.

For that reason, we propose using additional features that represent the communication structure rather than transferred data volume in bytes and packets. In addition to the flow count attribute for a certain aggregation key, with the second-level aggregation, we propose counting the appearances of all distinct elements of other identification attributes which are not used in the aggregation key.

As an example, a large number of distinct destination ports that some source IP address communicates with, labelled as $d[S]$, indicates a port scanning attack. Since these features represent the behaviour of network communications, we will call them *behaviour features*.

4.3. Slow initial stabilization

At the beginning of the baselining calculation, there is no history and previous data items to be used for the prediction. The entropy prediction in the next epoch can get the value of entropy in the current epoch, but more importantly, it is needed to properly estimate the standard deviation. A large initial standard deviation will cause a slow convergence process until it gets stabilized, which could keep some anomalies undetected (Figure 4).

Even more negative effect is produced when the standard deviation is too small, making a narrow threshold

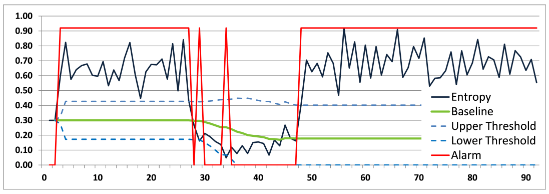

Figure 5: A narrow threshold margin with a small initial standard deviation (dataset CIC-IDS2017, trace Friday-DDOS, feature s[D]).

margin. Regular entropy values can easily fall out of the margin triggering a false positive alarm. More importantly, since the baselined values stay unchanged during alarms, this alarm could be locked for a longer period, which is shown in Figure 5.

The solution to this problem, which happens in practical implementation when the application is restarted, is to persist the last value of standard deviation calculated before the application restarted and reuse it as an initial value.

4.4. Entropy normalization

Feature normalization, such as linear transformation or z-score, is a commonly used approach to scale entropy values into a specific confidence interval to remove the bias and make data comparable [4]. In our research we use a linear transformation of entropy values, scaling entropy values relative to the thresholds, as follows:

$$\tilde{H}_n = \frac{H_n - \underline{T}_n}{\bar{T}_n - \underline{T}_n} \tag{11}$$

After normalization, the range of acceptable values is scaled into the interval $[0, 1]$, where the entropy values that are smaller than 0 or greater than 1 indicate anomalies. The greater the difference from the thresholds, the more reliable the alarm is.

4.5. Eliminating minor false positive alarms

To further distinguish minor and short-lasting anomalies from the more severe ones, illustrated in Figure 6, we define the anomaly score as a measure of deviation from the regular behaviour for each feature separately, taking into account the time period when the alarm is active. Obviously, higher deviation and longer alarm activation periods indicate greater anomalies. The rationale behind this is based on the fact that short spikes in entropy values could present acceptable data variations, due to which they are less important than deviations that last longer, even if they are less intensive.

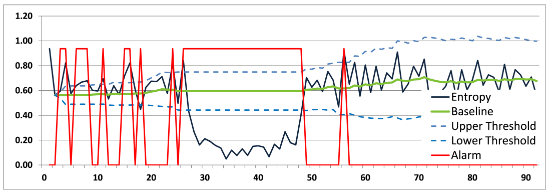

Figure 6: False positive alarms caused by minor entropy deviations.

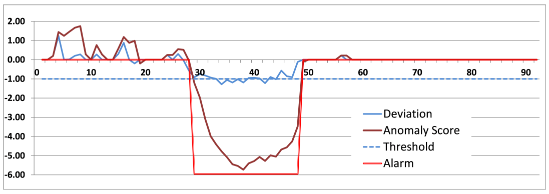

Figure 7: False positive alarms elimination by anomaly score.

To define the anomaly score, we first define the distance of the normalized entropy value from the thresholds (0 and 1) in epoch i, denoted as Δ_i, as follows:

$$\Delta_i = \begin{cases} \left|\tilde{H}_i\right|, & if\ \tilde{H}_i < 0 \\ \left|\tilde{H}_i - 1\right|, & if\ \tilde{H}_i > 1 \\ 0, & otherwise \end{cases} \tag{12}$$

Then, if the alarm is activated in epoch n, we define the anomaly score δ_n as a cumulative sum of the metric Δ_i in a sliding window during the last W' epochs or smaller, since the last alarm deactivation to avoid the influence of the previously triggered alarm. However, if the alarm is not active, a smaller sliding window is taken into account, W'', depending on how fast we would like to reset the anomaly score, where $W' = 1$ will reset it immediately.

$$\delta_n = \begin{cases} \sum_{i=1}^{W'} \Delta_{n-i+1} & if\ \Delta_n < 0 \\ \sum_{i=1}^{W''} \Delta_{n-i+1} & if\ \Delta_n = 0 \end{cases} \tag{13}$$

Figure 7 demonstrates the benefits of anomaly score usage in eliminating false positive alarms.

4.6. Flow partitioning

Detection performances of entropy-based approaches highly depend on the relative amount of anomalous activities in comparison to regular network behaviour for the observed feature. If the network is heavily loaded with regular traffic, the straightforward detection is limited only to highly intensive anomalies, while less aggressive malicious activities may remain undetected.

To address the above-mentioned issue, we propose the partitioning of network traffic into smaller subgroups and

Table 1
16 typical communication patterns.

S	s	D	d	Example of risk
1	1	1	1	Single flow
1	1	1	N	Port Scan
1	1	N	1	Network Scan
1	1	N	N	Diagonal Scan
1	N	1	1	Dictionary attack
1	N	1	N	Port Scan
1	N	N	1	Network Scan
1	N	N	N	Diagonal Scan
N	1	1	1	Amplification DDoS (DNS)
N	1	1	N	Amplification DDoS (NTP)
N	1	N	1	Multiple Network scan
N	1	N	N	Multiple Diagonal scan
N	N	1	1	SYN flooding
N	N	1	N	DDoS
N	N	N	1	Multiple DDoS
N	N	N	N	Multiple DDoS

apply a detection technique to each subgroup separately. This partitioning can be based on different criteria, such as the protocol type (TCP, UDP, ICMP), service type (DNS, email, web service, windows services etc.) or sub networking (user traffic, voice VLANs, data centre, branch offices etc.).

4.7. Communication patterns

Relying on the behaviour features rather than the volumetric features provides better capturing of different communication patterns. Considering the flow identification attributes and their cardinality in the communication (one or many occurrences), we developed 16 communication patterns. For example, the flows that present a port scan attack use a single source and destination IP addresses and many destination port numbers. Labelling with the cardinality one (“1”) or many (“N”), in order: source IP address, source port, destination IP address and destination port, namely (“Ss-Dd”), we can describe this port scan attack with the “11-1N” pattern. Consequently, if the source port number is randomly chosen with many occurrences (“N”), the communication pattern is “1N-1N”. Using this labelling convention, all 16 communication patterns, with the associated attacks are given in Table 1.

4.8. Anomaly modelling and classification

To analyze how different features are affected by different communication patterns, we have modelled the anomalies by injecting synthetically generated flows into a dataset of flows that correspond to normal traffic with no anomalies. We have obtained regular flow data from the public flow-based dataset CTU-13 [27], the trace “51“, by removing all flows associated with attacks and other background anomalies. The modelled anomalies were gradually increased to check the sensitivities of the features.

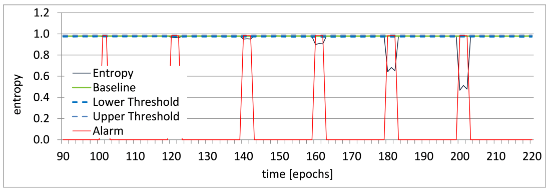

Figure 8: The entropy of the flow count feature aggregated by the source port (f[s]) caused by the anomalies following the N1-1N communication pattern.

Figure 8 shows the entropy drops caused by the anomalies following the “N1-1N” communication pattern which are detected with the flow count feature aggregated by the source port (f[s]).

The results obtained from the thorough analysis of which features are triggered by which anomaly type are summarized in Table 2. Even a brief overview of the table reveals that entropies behave differently for different communication patterns, while some of them are not affected by a particular anomaly at all (the empty cells in the table). More importantly, the ways how the entropies are affected by the modelled anomalies follow a very specific pattern.

It can be observed that the entropy drop (labelled ‘X’) occurs only when all features in the aggregation key have a single occurrence in the model (marked with ‘1’). For instance, aggregation by the source IP address causes an entropy drop only in the first 8 models, since a single host as a source of the anomaly greatly contributes to the calculated distributions.

In addition to anomaly detection, analyzing the triggered features can provide valuable information about the communication pattern of the anomaly, indicating the type of potential attack.

5. Implementation

When the research contributions were proven in the offline laboratory environment using the commonly used datasets, namely CTU-13 [27] and CIC-IDS2017 [28], the next step was to implement the concept for usage in real-life network communications. This section presents a developed architecture and the implementation details.

5.1. Architecture

A high-level architecture of the proposed methodology, illustrated in Figure 9, consists of the following main building blocks:

Table 2
Features affected by the communication patterns.

Model: Feature:	11-11	11-1N	11-N1	11-NN	1N-11	1N-1N	1N-N1	1N-NN	N1-11	N1-1N	N1-N1	N1-NN	NN-11	NN-1N	NN-N1	NN-NN
D[S]			X	X			X	X								
s[S]					X	X	X	X								
d[S]		X		X		X		X								
f[S]	X	X	X	X	X	X	X	X								
S[D]									X	X			X	X		
s[D]					X	X							X	X		
d[D]		X				X				X				X		
f[D]	X	X			X	X			X	X			X	X		
S[s]									X	X	X	X				
D[s]			X	X							X	X				
d[s]		X		X						X		X				
f[s]	X	X	X	X					X	X	X	X				
S[d]									X		X		X		X	
D[d]			X				X				X				X	
s[d]					X		X						X		X	
f[d]	X		X		X		X		X		X		X		X	
s[S.D]					X	X										
d[S.D]		X				X										
f[S.D]	X	X			X	X										
D[S.s]			X	X												
d[S.s]		X		X												
f[S.s]	X	X	X	X												
D[S.d]			X				X									
s[S.d]					X		X									
f[S.d]	X		X		X		X									
S[D.s]									X	X						
d[D.s]		X								X						
f[D.s]	X	X							X	X						
S[D.d]									X				X			
s[D.d]					X								X			
f[D.d]	X				X				X				X			
S[s.d]									X		X					
D[s.d]			X								X					
f[s.d]	X		X						X		X					
d[S.D.s]		X														
f[S.D.s]	X	X														
s[S.D.d]					X											
f[S.D.d]	X				X											
D[S.s.d]			X													
f[S.s.d]	X		X													
S[D.s.d]									X							
f[D.s.d]	X								X							
f[S.D.d.s]	X															

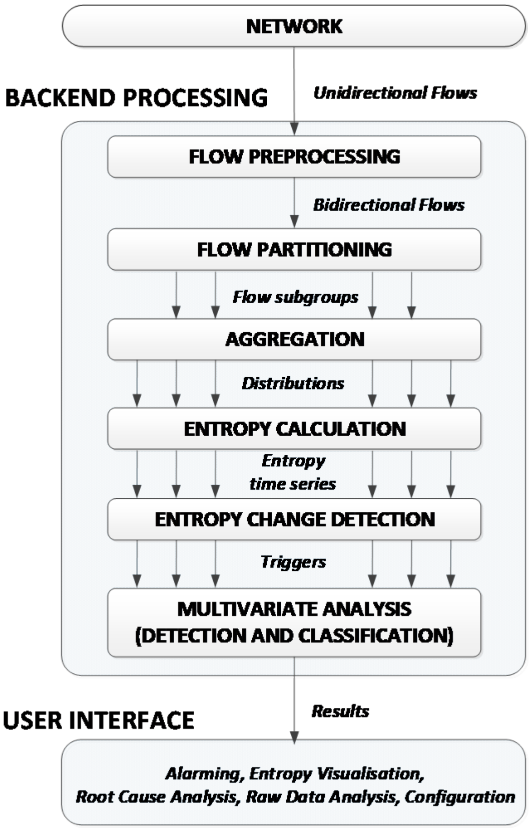

Figure 9: A high-level architecture.

- **Flow Preprocessing** – Two unidirectional flows from both directions between two peers are paired into a single record, the so-called bidirectional flow, which gives more information and ensures greater detection efficiency.
- **Flow partitioning** – Bidirectional flows are filtered by protocols, services or IP addresses, and divided into different sub-gropus, which are analyzed separately.
- **Aggregation** – Flow data during an epoch are aggregated based on the identification features, calculating additional behaviour features. The results are data distributions for each aggregation key and feature used.
- **Entropy calculation** – Entropy is calculated over each data distribution in each epoch, generating time series entropy values for every feature.
- **Entropy change detection** – A significant change in the entropy value indicates a change in network communication behaviour. The challenge is to accurately recognize changes resulting in anomalies and distinguish them from normal traffic variations.
- **Multivariate Analysis** – Alarms triggered by entropy changes are mutually analysed to provide a proper anomaly classification with higher detection accuracy.
- **User interface** – For practical implementation and usage in real-life networks, the obtained results need to be properly presented and managed, which includes simplified and meaningful visualisation, root cause analysis to extract anomalous data, efficient alarming, and root cause analysis with raw data inspection, including system configuration.

5.2. System implementation

The proposed architecture is implemented as a new module within the NetVizura solution, working in synergy with the NetVizura NetFlow Analyzer module [29]. Both modules can work on the same server as a monolith application, while in the case of higher traffic load they can be deployed on separate nodes, but still sharing the common user interface on the primary node. The backend application is developed in Java programming language, while the frontend user interface is developed in Javascript React framework combined with the legacy Google Web Toolkit elements.

Elasticsearch is used as a database for the entropy time-series data, top contributors in the data distributions used to calculate entropies, as well as raw flow data. As a non-SQL database, it is highly optimised and efficient for real-time data inserting at a high rate, as well as fast data retrieving.

A typical use case is when the user notices a single alarm containing the most relevant information, such as the anomaly class indicating the attack type. In a separate tab, shown in Figure 10, a user can inspect the entropy values in the recent period for a selected feature, while the individual elements that mostly contribute to the entropy changes are shown in a separate time chart. Click on any of these elements opens a new tab which shows raw data associated with the selected element. Users can further filter and aggregate the flow attributes and the resulting data are visualized at the throughput or volume chart (Figure 11).

The current version of the anomaly detection module is demonstrated in an operational environment (TRL-7 development level) achieving performances of 15 K flows/s on a server with 16 CPU cores and 32 GB RAM and using the Elasticsearch database on a separate server. Current development is focused on memory optimisation which appears to be the major bottleneck for a higher flow processing rate, expecting to achieve up to 50 K flows/s on a single server.

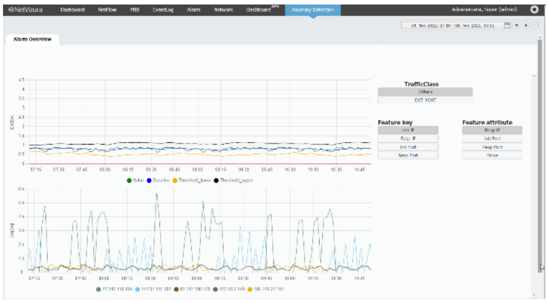

Figure 10: NetVizura Anomaly Detection - Entropy inspection.

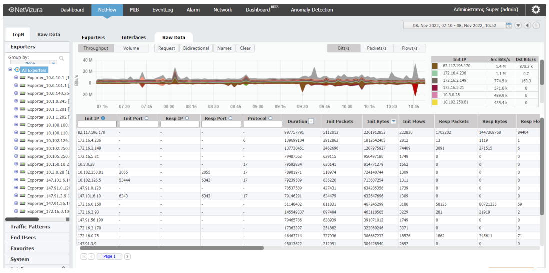

Figure 11: NetVizura Anomaly Detection - Raw data forensic.

6. Conclusion

In this paper, we have presented a comprehensive method for entropy-based network traffic anomaly detection and classification that relies on flow data. The method brings several novelties and improvements important for practical implementation.

Firstly, we developed a generalized concept of behaviour features based on the aggregation of the second degree, which better reflects the structure of network communications using only basic flow attributes. Based on these features, we have modelled 16 traffic patterns associated with anomalies that follow a wide range of security attacks. Secondly, we improved the entropy change detection process and reduced many false positive alarms by introducing the anomaly score metrics.

Thirdly, the comprehensive experiments have shown that different anomalies trigger alarms on different features, which can be interpreted as a characteristic signature that can be additionally used for anomaly classification. And finally, the applicability of the method is demonstrated through the implementation and real-time usage in real-life network environments.

The current work is oriented to performance optimization and improved usability, while further research is focused to the consolidation of multiple alarms using unsupervised machine learning.

Acknowledgments

The work presented in this paper has partially been funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia under the EUREKA project “Network Traffic Anomaly Detection system based on NetFlow data analysis – TRADE”, grant number E!13304 and the contract number 451-03-68/2022-14/200103.

References

[1] European Union Agency for Cybersecurity, ENISA threat landscape, 2022. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022.

[2] NexusGuard, DDoS threat report FHY 2022, 2022. URL: https://blog.nexusguard.com/threat-report/ddos-statistical-report-for-1hy-202.

[3] Truesec, An in-depth analysis of the cyber threat landscape, Truesec Threat Intelligence Report 2022, 2022. URL: https://www.truesec.com/hub/report/threat-intelligence-report-2022.

[4] N. Moustafa, J. Hu, J. Slay, A holistic review of network anomaly detection systems: A comprehensive survey, Journal of Network and Computer Applications 128 (2019) 33–55.

[5] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, H. Zhang, An empirical evaluation of entropy-based traffic anomaly detection, in: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, 2008, pp. 151–156.

[6] B. Tellenbach, M. Burkhart, D. Schatzmann, D. Gugelmann, D. Sornette, Accurate network anomaly classification with generalized entropy metrics, Computer Networks 55 (2011) 3485–3502.

[7] P. Bereziński, B. Jasiul, M. Szpyrka, An entropy-based network anomaly detection method, Entropy 17 (2015) 2367–2408.

[8] J. Ibrahim, S. Gajin, Entropy-based network traffic anomaly classification method resilient to deception, Computer Science and Information Systems 19 (2022) 87–116.

[9] S. Gajin, V. Timcenko, Comparison of entropy-based and machine learning approaches in intrusion detection, in: Proceedings of the 11th International Conference on Information Society and Technology ICIST 2021, 2021, pp. 113–118.

[10] N. Vichaidis, H. Tsunoda, G. M. Keeni, Analyzing darknet tcp traffic stability at different timescales, in: 2018 International Conference on Information Networking (ICOIN), IEEE, 2018, pp. 128–133.

[11] A. S. S. Navaz, V. Sangeetha, C. Prabhadevi, Entropy based Anomaly Detection System to Prevent

DDoS Attacks in Cloud, International Journal of Computer Applications 62 (2013) 42–47.

[12] S.-E. Benkabou, K. Benabdeslem, B. Canitia, Unsupervised outlier detection for time series by entropy and dynamic time warping, Knowledge and Information Systems 54 (2018) 463–486.

[13] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, B. Stiller, An overview of ip flow-based intrusion detection, IEEE communications surveys & tutorials 12 (2010) 343–356.

[14] B. Li, J. Springer, G. Bebis, M. H. Gunes, A survey of network flow applications, Journal of Network and Computer Applications 36 (2013) 567–581.

[15] İ. Özçelik, R. R. Brooks, Deceiving entropy based dos detection, Computers & Security 48 (2015) 234–245.

[16] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, ACM SIGCOMM computer communication review 35 (2005) 217–228.

[17] K. Xu, Z.-L. Zhang, S. Bhattacharyya, Internet traffic behavior profiling for network security monitoring, IEEE/ACM Transactions On Networking 16 (2008) 1241–1252.

[18] B. Claise, P. Trammell, B.and Aitken, Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, RFC 7011, IETF, 2013. URL: https://www.ietf.org/rfc/rfc7011.txt.

[19] C. E. Shannon, A mathematical theory of communication, The Bell system technical journal 27 (1948) 379–423.

[20] C. Tsallis, Possible generalization of boltzmann-gibbs statistics, Journal of statistical physics 52 (1988) 479–487.

[21] A. Rényi, et al., On measures of entropy and information, in: Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1, Berkeley, California, USA, 1961, pp. 547–561.

[22] A. Lawrance, P. Lewis, An exponential moving-average sequence and point process (ema1), Journal of Applied Probability 14 (1977) 98–113.

[23] A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, ACM SIGCOMM computer communication review 34 (2004) 219–230.

[24] P. Bojović, I. Bašičević, S. Ocovaj, M. Popović, A practical approach to detection of distributed denial-of-service attacks using a hybrid detection method, Computers & Electrical Engineering 73 (2019) 84–96.

[25] O. Joldzic, Z. Djuric, P. Vuletic, A transparent and scalable anomaly-based dos detection method, Computer Networks 104 (2016) 27–42.

[26] D. Rossi, S. Valenti, Fine-grained traffic classification with netflow data, in: Proceedings of the 6th international wireless communications and mobile computing conference, 2010, pp. 479–483.

[27] S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, computers & security 45 (2014) 100–123.

[28] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization., ICISSp 1 (2018) 108–116.

[29] NetVizura, Network monitoring solutions, 2022. URL: https://www.netvizura.com/.