INTERNATIONAL JOURNAL OF TECHNICAL AND NATURAL SCIENCES

Detecting Network Anomalies With Shannon Entropy: A Novel Approach to Cybersecurity

DOI: https://doi.org//10.69648/AOAL9594

International Journal of Natural and Technical Sciences (IJTNS), 2024; 4(1): 39-53

ijtns.ibupress.com

Online ISSN: 2671-3519

Application : 10.04.2024

Revision : 01.06.2024

Acceptance : 16.06.2024

Publication : 25.06.2024

Konicanin, V. (2024). Entropy based network anomaly detection. International Journal of Natural and Technical Sciences, 4(1), 39-54.

https://doi.org//10.69648/KJDP2991

Vahid Konicanin

International Balkan University, Makedonsko Kosovska Brigada, Skopje 1000, North Macedonia

https://orcid.org/0009-0002-4222-7484

We declare no conflicts of interest.

Correspondence concerning this article should be addressed to Vahid Konicanin

Email: vahid.k@ibu.edu.mk

Vahid Konicanin

Abstract

In an era of relentless cyber threats, the increasing complexity and volume of data significantly intensify the risk and impact of cybersecurity breaches. As organizations generate and store more data, the potential attack surface grows, providing more opportunities for malicious actors to exploit vulnerabilities. Consequently, there is a growing necessity for more advanced analytical techniques to effectively detect and mitigate these evolving threats. Shannon entropy, introduced by Claude Shannon in 1948, is a fundamental concept in information theory that measures the unpredictability or randomness of information. It serves as a primary tool for identifying unusual patterns within extensive datasets, offering a quantitative approach to detect anomalies

This paper explores the application of Shannon's Entropy to detect and prevent distributed denial-of-service (DDoS) attacks. Unlike traditional motif identification tools, which focus on recurring patterns within data, Shannon entropy provides a broader measure of randomness and can detect subtle variations that may indicate a security breach. By leveraging the entropy measure, cybersecurity systems can identify and respond to abnormal traffic patterns that signify a potential DDoS attack, thereby enhancing the robustness and reliability of data protection mechanisms.

BALKAN JOURNALS

Introduction – Shannon's Entropy

The concept of entropy is a fundamental principle that spans multiple scientific disciplines. Originating from the second law of thermodynamics, which states that the total entropy of an isolated system can never decrease over time, entropy has found a significant place in the domain of information theory. In information theory, it serves as a concise metric for randomness and disorder within a dataset or system. In statistical mechanics, the volume of the typical set of microstates is proportional to the exponential of the entropy, implying that higher entropy corresponds to a larger number of microstates that the system can occupy. Similarly, in information theory, the volume of the typical set of data sequences is related to the exponential of the Shannon entropy, indicating that higher entropy results in a larger set of probable sequences. For a probability distribution p (X = x) of a discrete random variable X, Shannon's entropy is defined as (Shannon, 1948):

$$H(X) = -\sum_{x \in \chi} p(x) log_2 p(x)$$

X is the feature that can take values {x1…xn} and p(x) is the probability mass function of outcome x [1]. The entropy H(X) is zero for p(x) = 0 because an impossible event carries no information. The entropy of a variable represents the "quantity of information" it holds, defined as the expected value of the self-information I(X;X)=-<$\log_2 p(x)$>, such that H(X)=I(X;X)=-$\sum p(x) \log_2 p(x)$.

However, this quantity doesn't solely depend on how many possible states the variable might have but also on the probability distribution of these states. For example, if all outcomes are equally likely, the entropy is maximized, reflecting maximum uncertainty. Conversely, if some outcomes are much more likely than others, the entropy is lower, indicating less uncertainty. In the same vein, the informational content of an email isn't measured merely by the count of words or the vocabulary used. Informally, the level of information in an email correlate to the degree of "unexpectedness" in its message. For instance, a repetitive email offers no new information. Conversely, an email revealing the unexpected outcome of a highly anticipated sports final is extremely informative. The informative value in a variable links to the level of surprise we experience upon learning its current state, which is directly influenced by the probability distribution of the outcomes. If an event is highly unlikely (low probability), the surprise and thus the information content is higher, contributing more to the overall entropy H(X). Therefore, the entropy reflects not just the number of possible states, but also how those states are probabilistically distributed.

Shannon's entropy metric quantifies the self-information within a variable, thus laying the groundwork for a systematic understanding of the concept of information. Self-information, denoted as `I(x)` is defined as I(x)=-log p(x), where p(x) is the probability of a specific outcome x. It measures the amount of surprise or information content associated with the outcome x. The quantity of information typically relates to the complexity involved in storing and conveying it. Take a coin flip, for example. The result can be encoded within a binary variable that can hold a 0 or a 1, indicating heads or tails. This variable directly reflects the raw outcome of the flip, which can easily be represented by a single bit, the fundamental unit in digital storage and communication technologies. However, this bit only contains the immediate result, not the deeper details such as whether the coin has a bias and to what degree, if any.

In the case of an experiment with two possible outcomes, X=0 with probability p and X=1 with probability `1−p`, the entropy H(X) is calculated as:

$$"H(X) = -[p\ log\ p + (1-p)\ \ log(1-p)]"$$

The entropy H(X) is maximum when p=0.5. At this point, H(X)=1 bit. This indicates that the uncertainty or unpredictability is highest when the coin is fair, as both outcomes are equally likely; therefore, H(X) (which equals I(X;X)) represents the information gained if the discrete random variable is completely described.

Relating this experiment to self-information, when p=0.5, each outcome (heads or tails) has a self-information of –log 0.5 = 1 bit. This maximum self-information aligns with the maximum entropy, reflecting the highest level of surprise and uncertainty in the result of a fair coin flip. When the coin is biased (i.e., p is not 0.5), the entropy decreases, indicating less uncertainty and thus less information content in each outcome.

Using Shannon's Entropy to Analyze Network Packets

Shannon's entropy can be a valuable tool in network security by evaluating the randomness of network traffic. When applied to the analysis of network packets, entropy can help in identifying patterns that deviate from what is considered normal behavior, effectively signaling potential security threats such as malware communication, data exfiltration, or Distributed Denial of Service (DDoS) attacks. For instance, in a network system, the packets being transferred should follow a relatively predictable distribution if the system is uncompromised and functioning

normally. However, if an attacker injects malicious packets or a large amount of nonsensical traffic to disrupt service, the entropy of the network traffic can increase out of a given threshold due to the added unpredictability. By monitoring changes in entropy in real time, security systems can alert administrators to anomalous behavior that may indicate a security breach. (Bakhare & Mohod, 2024; Scalabrin et al., 2017).

In typical network traffic, packets often consist of structured data conforming to various protocols. The entropy of such packets would generally fall within a predictable range, reflecting the regularity of the packet contents. These contents might include headers, session information, and payloads that, while varying in content, follow certain structural patterns inherent to the communication protocols and services in use (Gu et al., 2005).

However, in the case of network intrusions or non-standard activities such as malware communication or data exfiltration, the packet contents might deviate significantly from the norm, either exhibiting abnormally low or high entropy. For instance, a packet carrying a simple, repetitive payload generated by a network attack tool might exhibit a change in entropy due to the altered probability distribution (Berezinski et al., 2015). Similarly, an encrypted message used by malware to communicate undetected might cause a significant fluctuation in entropy due to its randomized appearance. Hence, it is not merely the presence of low or high entropy that signals an anomaly but rather the change in the probability distribution, which alters the Shannon entropy, indicating irregular network activity.

The practical deployment of this entropy-based strategy would involve processing packets in real-time or near-real-time, calculating their entropy, and comparing this value against the predefined thresholds. Any detections outside the bounds would trigger alerts for further inspection. Over time, the thresholds may need to be adjusted to accommodate for changes in normal traffic patterns, ensuring the system's accuracy and reducing false positives or negatives. By integrating the entropy-based analysis with other context-aware detection mechanisms, it is possible to create a more robust and reliable anomaly detection system (Berezinski et al., 2015). It must be noted, though, that sophisticated attackers may craft their network activities to mimic typical entropy patterns, thus reinforcing the need for layered defense strategies in network security (Mukherjee et al., 2020).

Beyond Shannon's entropy, which has its roots in the discrete binary context of messages and signals, there are various other types of entropy that play significant

roles across different fields, each providing a distinct perspective on disorder, uncertainty, and informational content.

Related Works

Generalized Entropies: Parameterization and Adjustability in Anomaly Detection

Generalized measures of entropy have emerged recently as effective tools within the realm of network anomaly detection. Research by Eimann (2008), indicates favorable results when utilizing T-entropy to detect intrusions through an analysis of network packets. This form of entropy relies on a metric known as T-complexity, which quantifies string complexity based on the minimum steps required to construct a specific string. Unlike the probabilistic foundation of traditional entropy, where frequency-derived probabilities can be interchangeable, the complexity approach necessitates the preservation of sequence order. The compression algorithm's output length offers an approximation of the string's complexity, which then serves as an entropy proxy. Considering its emphasis on event sequence, this method is apt for detailed network data scrutiny methods, such as complete or header-only packet inspection.

Tsallis Entropy: Determining the Optimal α Parameter for DoS Attack Detection

Beyond T-entropy, entropy generalizations that allow parameterization have shown considerable potential. These include Tsallis entropy (2011) and Renyi entropy (1970), which afford control over the emphasis between a distribution's core and its tail. Tsallis entropy is a generalization of the standard Boltzmann-Gibbs entropy. It introduces a parameter q, which allows the entropy to be adjusted to give different weight to different parts of the probability distribution. An α value around 0.9 means the detection method emphasizes the distribution's primary mass. This approach is effective because DoS attacks typically generate a high volume of traffic, concentrating the distribution's primary mass. Rényi entropy is another generalization of entropy, parameterized by α. It provides a spectrum of entropy measures, which include the Shannon entropy as a special case. An α value around 0.5 focuses less on the distribution's core and more on the tail, which is suitable for detecting port scan anomalies that may not dominate the primary mass of the distribution but rather affect its tail.

By tuning the parameter q (or α), both Tsallis and Rényi entropies allow for more flexible and targeted analysis of probability distributions. This flexibility can be particularly useful in cybersecurity applications like DoS attack detection and port scan anomaly detection, where different types of anomalies can be more effectively identified by adjusting the entropy parameters.

Entropy-Based Attribute Selection for Enhanced Intrusion Detection Systems

Comparative analyses of Shannon, Renyi, and Tsallis entropies in attribute selection to improve the detection abilities of decision tree and k-means classifiers were conducted by Lima (2012). Through experiments, it was found that models developed using a minimized subset of attributes can perform on par with, or even exceed, models utilizing the full attribute set, particularly in relation to DoS and scanning attack scenarios. Specifically, for the DoS category, the research articulated how smaller attribute sets could be equally if not more effective.

Proof of Concept – Calculating Entropy, Normalization, Thresholds

Shannon's entropy can also be applied in the analysis of packet payloads on which we will be focusing. Greater randomness in the payload may be an indication of encryption or obfuscation used by attackers to hide their data from standard detection techniques. Conversely, a low entropy may indicate a potential spam campaign, as such messages often contain repetitive text and thus are more predictable (Berezinski et al., 2015).

In the following code, "freqs" is defined as a list comprehension. It's a list that contains the frequencies of each unique character in the input string. The frequency of a character is calculated by counting how many times that character appears in the string (string.count(char)) and then dividing by the total length of the string (len(string)), resulting in a float that represents the proportion of the string that is made up by that character. The Shannon entropy is then calculated as the negative sum of each frequency multiplied by the logarithm base 2 of that frequency, which gives a measure of the uncertainty or randomness of the string based on its probability distribution.

```python
def calculate_entropy(string):

    freqs = [float(string.count(char)) / len(string) for char in set(string)]

    shannon_entropy = -sum([freq * math.log2(freq) for freq in freqs])

    return shannon_entropy
```

Figure 1 [2]

The function from the code in Figure 1 calculates the Shannon entropy of a given hex input (converted to string). As we've established, entropy is a measure of unpredictability or self-information content, and in the realm of information theory, Shannon's formula is pivotal. The function first computes the frequency of each unique character in the string, and then applies Shannon's entropy formula. This formula inherently penalizes predictability in the dataset, resulting in higher entropy values for more random or less predictable data.

Normalized entropy in the context of the following Python code is a measure of the randomness or unpredictability of information in a given network packet (hexadecimal representation), adjusted relative to the maximum possible entropy for that string given the number of unique symbols it contains. It ranges from 0 to 1, where:

- 0 indicates no entropy (complete predictability)
- 1 indicates maximum entropy (complete randomness/no predictability)

The normalization is done to compare entropy values across different datasets or strings with various numbers of unique symbols, giving a relative measure that accounts for these differences. The maximum entropy occurs when all symbols in the dataset are equally likely. This scenario represents the highest level of uncertainty because there is no preference for any particular symbol.

```python
def calculate_normalized_entropy(string):
    freqs = [float(string.count(char)) / len(string) for char in set(string)]
    entropy = -sum([freq * math.log2(freq) for freq in freqs])

    max_entropy = math.log2(len(set(string)))
    normalized_entropy = entropy / max_entropy

    return normalized_entropy
```

Figure 2 [1]

Following a similar structure to the previous function, this one computes the entropy of a string input but goes a step further by normalizing this value. Normalization is accomplished by diving the Shannon entropy by the maximum possible entropy, which is the logarithm base 2 of the cardinality of the set containing unique characters in the input string (packet dump). The normalized entropy is a relative measure, enabling better comparison across different data sets by accounting for differences in size and composition.

From an information theory standpoint, the average normalized entropy of a set of network packets can provide insight into the overall predictability and uniformity of the data being transmitted.

The interpretation of the average normalized entropy in the context of analyzing network packets from a .pcap file can be as follows:

- Close to 0: If the average normalized entropy is closer to 0, it suggests there is low randomness across the payloads of the packets analyzed. This homogeneous behavior can be typical in scenarios with lots of repetitive data, such as a spam campaign, a coordinated transfer of a large file with redundant content, or simple protocol handshakes that involve repeated predictable packets.
- Close to 1: An average normalized entropy closer to 1 indicates that the payloads exhibit a high degree of randomness. This might be characteristic of encrypted or compressed data, which ideally look like random noise in their byte distributions. In security, this could signify that the traffic contains encrypted communication, used legitimately or potentially by attackers to hide malicious payloads.
- Moderate Values: Values in between could suggest a mix of predictable and unpredictable data, or data that is encoded in a way that does not reach the extreme of randomness exhibited by encryption but is less predictable than plaintext traffic.

```
def process_pcap(file_path, num_packets):
    entropy_info = {}
    entropy_values = []
    entropy_sum = 0
    with open(file_path, 'rb') as pcap_file:
        pcap = dpkt.pcap.Reader(pcap_file)
        for i, (timestamp, packet) in enumerate(pcap):
            if i >= num_packets:
                break
            packet_string = ''.join(hexdump.dump(packet).split()[1:])
            eth = dpkt.ethernet.Ethernet(packet)
            if isinstance(eth.data, dpkt.ip.IP):
                ip_address = socket.inet_ntoa(eth.data.src)
                normalized_entropy = calculate_normalized_entropy(packet_string)
                entropy_sum += normalized_entropy
                if ip_address not in entropy_info:
                    entropy_info[ip_address] = []

                entropy_info[ip_address].append(normalized_entropy)
                entropy_values.append(normalized_entropy)
    global_avg_normalized_entropy = statistics.mean(entropy_values)
    global_std_dev = statistics.stdev(entropy_values)
    outlier_threshold = global_avg_normalized_entropy + global_std_dev
```

Figure 3 [1]

Acting as the core analytic mechanism, this function processes the packets within a "pcap" file. After opening the file, the function iterates through each packet up to a predetermined number to manage computation resources. For each processed packet, the script extracts the raw packet data and converts it to a string with hexadecimal representation. By doing this, the script can obtain the source IP address of the packet using "dpkt" python library parsing.

Subsequently, the function leverages a function "calculate_normalized_entropy" to compute the normalized entropy of the packet and stores this metric in association with the corresponding IP address. To manage the collected data efficiently, the script utilizes dictionaries and lists, grouping entropy values by IP and maintaining a comprehensive list of all entropy values across packets.

Following the iterative packet processing, the function computes the average normalized entropy for the subset of packets processed. It also calculates the global average normalized entropy and the standard deviation of these entropy values across all processed packets to establish a reference for normal data patterns.

With the insights gained from these statistics, the function calculates a threshold for identifying outliers; any IP addresses with entropy values that exceed the sum of the global average normalized entropy and the standard deviation are considered suspicious. These high entropy outliers might be highlighting non-standard behavior, such as potential security threats or data exfiltration attempts. Thus, by accounting for the average and standard deviation in identifying outliers instead of a static predefined threshold, the script exhibits added sophistication in detecting anomalies, adapting to the variability observed in the dataset.

Detection of Distributed Denial of Service (DDoS) Attack Using Entropy

Recently, the gaming server infrastructure that I manage was subjected to a Distributed Denial of Service (DDoS) attack, which presented considerable challenges in the identification of malicious traffic sources. The attackers effectively "masked" the attack by attacking the server from within the server (instead of DDoS-ing externally using a botnet or other more "traditional" methods), thereby rendering conventional identification methodologies inadequate in isolating the source of the attack.

In typical network settings, especially in environments with well-defined and structured traffic patterns, network packets often exhibit a degree of regularity and predictability. This is particularly true for gaming servers, where the interac-

tions often follow predefined protocols and patterns. The data packets generated by games are usually of consistent sizes and occur at expected intervals. This reflects routine player actions, standard game-state updates, and habitual communication between clients and the server.

For instance, in the context of a multiplayer online game, player movements, actions, game physics updates, and chat messages lead to a continuous, but predictable stream of network traffic. The packets involved in these transactions tend to have standardized headers and payload structures, resulting in low entropy values under normal conditions

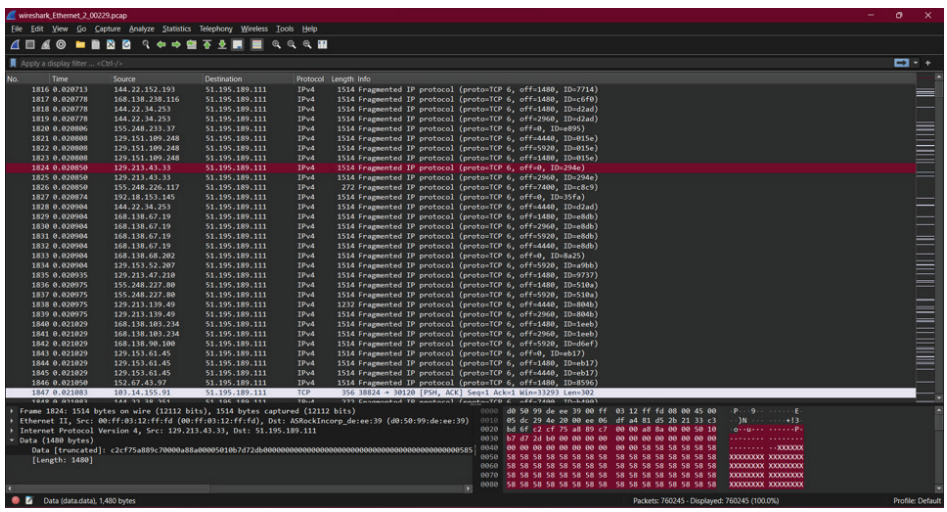

Figure 4: Captured Packet Dump in Wireshark

In an attempt to address this issue, we employed a Python script developed for the purpose of analyzing the traffic captured during the incursion, which was stored in the form of a .pcap" file with TCP packet data (as shown in Figure 4).

The core strategy involved was measuring the entropy of packet data to uncover patterns characteristic of DDoS traffic. Entropy, a concept derived from information theory, quantifies the unpredictability or randomness in a dataset. In the context of network traffic, higher levels of entropy are often indicative of attack patterns due to non-standard distribution of packet sizes and frequencies.

The statistical significance of a large value of average normalized entropy in the context of detecting DDoS attacks boils down to its deviation from what is considered "normal" behavior for network traffic. "Normal" behavior would typically be

characterized by a consistent traffic pattern that doesn't fluctuate widely in terms of packet size distribution and frequency, which translates to relatively stable entropy values.

```
IP Address: 158.101.31.194, Average Normalized Entropy: 0.3471
IP Address: 144.22.252.76, Average Normalized Entropy: 0.3156
IP Address: 129.153.52.207, Average Normalized Entropy: 0.3569
IP Address: 155.248.233.37, Average Normalized Entropy: 0.3517
IP Address: 140.238.146.218, Average Normalized Entropy: 0.3478
IP Address: 129.151.125.171, Average Normalized Entropy: 0.3484
IP Address: 192.18.144.166, Average Normalized Entropy: 0.3373
IP Address: 150.136.162.49, Average Normalized Entropy: 0.3373
IP Address: 193.123.107.230, Average Normalized Entropy: 0.3434
IP Address: 51.195.189.111, Average Normalized Entropy: 0.8454
IP Address: 168.138.70.209, Average Normalized Entropy: 0.3427
IP Address: 103.14.155.188, Average Normalized Entropy: 0.9527
IP Address: 72.219.87.116, Average Normalized Entropy: 0.9325
IP Address: 144.22.242.11, Average Normalized Entropy: 0.3157
IP Address: 173.47.119.20, Average Normalized Entropy: 0.9066
IP Address: 71.60.151.152, Average Normalized Entropy: 0.9268
IP Address: 168.138.127.158, Average Normalized Entropy: 0.8892
```

Figure 5: Output of a script that analyzes packet dumps [1]

Establishing a Baseline

The average normalized entropy for the analyzed packets, during a period of normal operation without a DDoS attack, was calculated to be "0.3701501927904838"

Variance, Standard Deviation, Threshold

IP addresses with abnormally high normalized entropy values were identified as outliers. An outlier threshold was dynamically set as the sum of the global average normalized entropy and the standard deviation of normalized entropy values. IP addresses with average normalized entropy exceeding this dynamically determined threshold - indicative of unusual and potentially malicious activity - were highlighted in red (as presented in Figure 5). These higher entropy values suggest a notable deviation from typical network traffic patterns and could be indicative of a DDoS attack methodology. Using entropy analysis for the detection of DDoS attacks provided a measurable metric for identifying abnormal traffic patterns.

Enhanced Automated Response Mechanism

Building upon the results from the initial baseline analysis, we have refined our Python script to include an automated response mechanism. The script now continually assesses the average normalized entropy for each IP address. If the entropy exceeds the dynamic threshold associated with abnormal behavior, it automatically triggers a block against the offending IP address. This effectively prevents further access to the game server from that IP, serving as a preemptive defense against potential DDoS attacks. The inclusion of this self-actuating feature considerably narrows the window of opportunity for attackers, potentially diminishing the severity or entirely averting the damages inflicted by a DDoS attack.

The integration of this feature represents a shift from reactive analysis to active prevention. By regularly capturing snippets of network traffic and calculating the entropy in real-time or near-real-time, this tool can effectively monitor for unusual patterns that indicate a DDoS attack is in progress or imminent. This automated and continuous vigilance ensures that network security measures can adapt swiftly to potential threats, allowing for the dynamic adjustment of defenses and the pre-emptive blocking of malicious actors. In essence, such a tool transcends traditional security measures by adding a layer of intelligence and rapid-response capability to the network's arsenal, providing a robust safeguard that can adapt to the evolving landscape of cyber threats.

Conclusion

As we navigate the ever-evolving threat landscape of cyberspace, it becomes increasingly clear that traditional cybersecurity measures must be bolstered by more sophisticated approaches to effectively combat modern strategies employed by malicious actors. The application of Shannon entropy in anomaly detection systems, as evidenced by the research conducted and the practical implementation within a gaming server infrastructure, undeniably demonstrates the capacity of this statistical measure to serve as a reliable indicator of abnormal network behavior, specifically in the identification and prevention of Distributed Denial of Service (DDoS) attacks.

Throughout the progression of this research paper, I have explicated the theoretical underpinnings of Shannon's Entropy and its intrinsic value in discerning patterns indicative of cyber threats from within extensive datasets. Empirical analysis, facilitated by the development of a heuristic Python script, validated the hypothesis

that entropy can unveil unusual traffic patterns, which are otherwise difficult to detect using conventional security tools. The successful isolation and subsequent blocking of IP addresses with abnormally high entropy values endorse the premise that entropy is not merely a diagnostic metric but a critical component in a dynamic cyber defense strategy.

The automated response mechanism integrated into our Python script epitomizes a significant leap from passive detection to active and real-time mitigation of security incidents. By continuously scrutinizing network traffic for entropy deviations, we have established a responsive and adaptive security posture capable of pre-emptively neutralizing potential threats. The immeasurable benefit of reducing the reaction timeframe in the face of an attack cannot be overstated. This agility ensures that the security of networked systems can pivot swiftly and effectively in response to the complex array of modern cyber incursions.

In conclusion, the evolving threat landscape of cyberspace necessitates the enhancement of traditional cybersecurity measures with sophisticated approaches. The application of Shannon entropy in anomaly detection systems, as demonstrated by this research, highlights its effectiveness in identifying and preventing Distributed Denial of Service (DDoS) attacks. Throughout this paper, the theoretical underpinnings of Shannon's entropy and its practical implementation within a gaming server infrastructure have been explored. The empirical analysis, facilitated by a heuristic Python script, validated the hypothesis that entropy can unveil unusual traffic patterns that are difficult to detect using conventional security tools. The successful isolation and subsequent blocking of IP addresses with abnormally high entropy values endorse the premise that entropy is not merely a diagnostic metric but a critical component in a dynamic cyber defense strategy.

References

Bakhare S., Mohod S.W., (2024). A Review on Real-Time Network Traffic Monitoring and Anomaly Detection System : A Comprehensive Study with User-Friendly Interface and Historical Analysis Capabilities. International Journal of Scientific Research, Vol. 11 No. 3 (2024): May-June.

Berezinski, P., Jasiul, B., Szpyrka, M. (2015). An Entropy-Based Network Anomaly Detection Method, Article

Eimann, R. (2008). Network event detection with entropy measures (Ph.D. thesis). University of Auckland, Auckland, New Zealand.

GitHub Gist. (n.d.). Source code, entro.py – Simplified network packet analysis script. Full source code available at: https://gist.github.com/jinnosux/63160c7cf9d929f7eb9ce0221917b345

Gu, Y., McCallum, A., & Towsley, D. (2005). Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation. Proceedings of the 5th Conference on Internet Measurement, 19-21, 2005

Lima, C. F. L., de Assis, F. M., & de Souza, C. P. (2012). A comparative study of use of Shannon, Rényi, and Tsallis entropy for attribute selecting in network intrusion detection. In Proceedings of the 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'12), Natal, Brazil, 29-31 August 2012 (pp. 492–501).

Mukherjee, S., Heberlein, L. T., & Levitt, K. N. (2020). Network Intrusion Detection. IEEE Network, 8(3), 26-41.

Renyi, A. (1970). Probability theory. Amsterdam, The Netherlands: North-Holland.

M. Scalabrin, M. Gadaleta, R. Bonetto and M. Rossi, "A Bayesian forecasting and anomaly detection framework for vehicular monitoring networks," *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, Tokyo, Japan, 2017, pp. 1-6, doi: 10.1109/MLSP.2017.8168151.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3), 379-423.

Scalabrin, M., Gadaleta, M., Bonetto, R. & Rossi, M. (2017). A Bayesian forecasting and anomaly detection framework for vehicular monitoring networks. *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, Tokyo, Japan, 2017, pp. 1-6, doi: 10.1109/MLSP.2017.8168151.

Tsallis, C. (2011). The nonadditive entropy Sq and its applications in physics and elsewhere: Some remarks. Entropy, 13, 1765–1804.