Real-time Network Traffic Analysis and Feature Extraction

Computer Systems and Networks (448.015)

January, 2025

Riccardo Baljak
Ismar Nurdnović
Javier Nieto Castaño

The Packet Inspectors

***Abstract*—This paper is about real-time network traffic analysis and feature extraction. Its first part covers the theoretical background needed for understanding the topic. Real-time analysis there is a demand nowadays, concerning the new concepts of cloud (fog) computing and the internet of things. Breaking down the network traffic into reasonable chunks is essential, because otherwise it would be unmanageable to deal with it. Also network threats, for example (distributed) denial of service ([D]DoS) attacks, make a severe impact on network security and therefore need to be detected quickly. Packets are just one key concept in network traffic analysis. They contain information, for example about the source, the destination or the protocol. Inspecting the packets using flow-based techniques analyze cumulative communication flows to identify patterns and anomalies. Machine learning algorithms or edge computing are modern features used for such inspections. Even though real-time traffic analysis has a bunch of advantages, the big data flow running through the modern networks (network infrastructure) at high speeds are a big downside. Older tools tend to brawl with all that and reach their limits. This leads to the invention of new methods. Mitigation of malicious software or activities is just one application of real-time traffic analysis, while AI-driven development is coming up in the future. The second part focuses on the findings, which are about the practical part. There were a feature extractions done in the code. These were divided into Ethernet, IP and TCP/UDP. Ethernet frames just contain the source and destination MAC address. A source and destination address can be found with IP as well and also the version number of the IP and the protocol used (TCP, UDP). TCP/UDP headers both contain a source and destination port and a checksum over the message. UDP only has one more feature, the length of the data and header while TCP has a lot more features, only one being the flag(s).**
To capture anomalies there are way complex algorithms that use machine learning, some basic filters which just look at the packet size or if the dns has some strange pattern in it. Also there is a web page where user uploads some malicious ips so other people can be aware of it. Finally, the challenges are tangled. Due to the rather short code not many challenges appeared and the ones that appeared where due to installation and privilege issues.

I. Introduction

The immense growth of the Internet has made real-time network traffic analysis a cornerstone of effective network management and cybersecurity. Real-time network traffic analysis alludes to immediate monitoring and evaluation of data packets as they travel through a network, allowing the identification of patterns, detection of anomalies, and rapid response to emerging threats. This capability is particularly crucial in today's modern trends, unlike older methods, which analyze data after the fact. [Bar+20]

In this work, we seek to address these challenges by exploring the latest advances in real-time network traffic analysis and feature extraction. We examine small methodologies for processing and interpreting traffic data.

To summarize, we make the following contributions:

- We explain and analyze Real-time Network and traffic analysis and feature extraction.
- We explore and demonstrate different methods of Real-time Network and Traffic Analysis.

Outline. Section 2 provides background. In Section 3, we analyze and explore different types of Real-Time Network analysis. In Section 4 we discuss encountered challenges. And finally, we conclude in Section 5.

II. Background

This section provides a theoretical background relevant to real-time network traffic analysis. This includes the current need for proper real-time network analysis, the key concepts, challenges, and future directions of real-time network traffic analysis.

A. Importance of Real-time Analysis

The increasing integration of cloud computing, Internet of Things (IoT) devices, and 5G networks has exponentially expanded the volume and variety of network traffic. As the network infrastructure grows in scale and complexity, traditional approaches of retrospectively analyzing traffic using batch processing techniques become inadequate. We have seen cybersecurity threats, such as Distributed Denial of Service (DDoS) attacks [DM04] and advanced persistent threats (APTs) [DM04], which require immediate detection and response to minimize damage. Real-time analysis enables organizations to act on potential threats immediately, by doing so, it reduces the window of vulnerability and enhances the resilience of critical infrastructure [Som+17].

B. Key Concepts in Traffic Analysis

Network traffic analysis involves capturing, inspecting, and interpreting the data packets transmitted over a network. Each packet contains headers and payloads that contain essential information about the source, destination, protocol, and content of the communication. Real-time traffic analysis systems leverage packet inspection, flow-based monitoring, and statistical modeling to derive actionable insights. Packet inspection examines individual packets, while flow-based techniques analyze aggregated communication flows to identify patterns and anomalies [DS12]. To achieve real-time performance, modern systems integrate advanced technologies such as deep learning, edge computing, and big data processing frameworks. Machine learning algorithms could be used to classify traffic patterns and detect anomalies in real-time, often using features such as packet size distributions, flow durations, and protocol types [NA08]. Edge computing enhances real-time capabilities by processing data closer to the source, thereby reducing latency and bandwidth consumption [Shi+16].

C. Challenges in Real-time Traffic Analysis

Despite all of its advantages, real-time network traffic analysis poses significant technical and operational challenges. The sheer volume of data generated by high-speed networks requires highly upgradeable and efficient frameworks. Traditional tools often struggle to keep up with the data rates observed in modern networks, which can exceed several terabits per second [Tro19]. Another additional variable, is the increasing use of encryption, such as HTTPS and VPNs, complicating the analysis by obscuring packet content. This type of analysis must rely on metadata and behavioral analysis to infer patterns in encrypted traffic [AM20]. Privacy concerns also emerge as a critical issue, as traffic analysis involves examining potentially sensitive data. Regulatory frameworks, such as the EU General Data Protection Regulation (GDPR), mandate strict controls over data access and processing, which can limit the analysis [VB17]. Balancing security needs with privacy requirements is a constant challenge for network maintainers and researchers alike [Cav10].

D. Applications and Future Directions

Real-time traffic analysis is applied across various domains. For example, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) rely on real-time analysis

to identify and mitigate malicious activities before they impact operations [Sna+91]. Similarly, traffic optimization tools use real-time data to balance loads, reduce congestion, and improve user experiences [Xu+11]. The integration of AI-driven automation promises to advance the field further. Additionally, the adoption of decentralized architectures, such as blockchain, may enhance the security and reliability of traffic analysis processes [Dor+17].

III. Findings

The code, which is handed out separately, mainly uses the `pyshark` library among others. It's a `tshark` wrapper for `Python3`. In order for it to work, `Python` version ≤ 3.13 must be used at the moment, as some internal dependencies rely on it. After defining the interface for the so-called `liveCapture pcap reader`, for example the WiFi adapter, one can start sniffing packets continuously, either running forever or defining a maximum number of packets.

A. Packet Number and Timestamp

The first two things we extract are the packet number and the timestamp. They are the most basic information available for each packet. While the packet number is just a consecutive number starting from 1, the timestamp is the actual sniff time. Both therefore are not a part of a packet itself, but rather a part of the framework. The packet number resets each (re-)run of the program.

B. Ethernet

If the packet contains the string `eth`, we know that we can obtain the Ethernet parameters of it. These are the source and destination `MAC` address. There is no more interesting information on this layer we can sniff.

C. Internet Protocol – IP

On the next layer of the TCP/IP stack, the internet layer, we can sniff information, i.e., extract the features of the Internet Protocol. This is only done if `ip` is in the packet. First we distinguish between IPv**4** and IPv**6** to get access to the right features. These include the protocol version of the Internet Protocol and the name of the underlying protocol (which is actually on the transport layer) as well as it's number (defined by the iana). After that, we extract the source and destination IP address.

D. TCP/UDP

The transport layer includes the two protocols TCP and UDP. Both have a source and destination port we can extract, as well as the checksum. TCP has a flag and UDP has a length which can be sniffed too.

TCP: Flags could be: `NS`, `CWR`, `ECE`, `URG`, `ACK`, `PSH`, `RST`, `SYN`, `FIN`, or a combination of them. First `SYN` is sent from the sender to the receiver, the latter then replies with `SYN/ACK`, the sender then replies with `ACK` and starts sending data. There are even more features, like the urgent pointer, to extract, but these are not covered by the code.

UDP: This protocol does not have flags, but a length field in the header which contains the length of the header and data. Unlike TCP, there are no more fields in the header. UDP is less reliable, but faster than TCP.

E. Anomalies

Unusual or unexpected patterns that deviate from normal behavior can indicate security threats or performance behavior. That is why detecting anomalies is very critical to keep safe your data. They are normally detected using real-time algorithms that use machine learning, continuously monitoring the network traffic.

IV. Challenges

There were a few challenges faced during the development of the short code. The installation itself should start with a Python Virtual Environment (`venv`), in order to prevent interference with packages with could be globally installed. Just installing `pyshark` with Python's package manager `pip` is not enough: `tshark` has to be installed as well. The latter is not a Python package but a command line interface for the

well-known packet sniffer WireShark. Another issue faced was that `tshark`, more precisely `tshark -D` which is necessary for the execution of the `pyshark` script, required `root` privileges. After resolving this issues, the development could start.

V. Conclusion

In this work, we explained what Real-time Network and traffic analysis and feature extraction are. We discussed the importance of security and compliance in these real-time systems. Additionally, we reviewed various approaches to real-time traffic monitoring, classification, and anomaly detection, and presented the challenges we faced along the way.

References

[AM20] J. P. Anderson and D. McGrew. “Encrypted traffic analysis: Techniques and challenges.” In: *Journal of Cybersecurity* (2020).

[Bar+20] M. Barabas et al. “Real-time network traffic monitoring and analysis: A survey.” In: *Journal of Network and Computer Applications* (2020).

[Cav10] A. Cavoukian. *Privacy by Design: The 7 Foundational Principles*. Information and Privacy Commissioner of Ontario, Canada. 2010.

[DM04] C. Douligeris and A. Mitrokotsa. “DDoS attacks and defense mechanisms: classification and state-of-the-art.” In: *Computer Networks* (2004).

[Dor+17] A. Dorri et al. “Blockchain for IoT security and privacy: The case study of a smart home.” In: *IEEE PerCom Workshops* (2017).

[DS12] F. Dressler and C. Sommer. “Traffic analysis and pattern recognition in communication networks.” In: *IEEE Communications Magazine* (2012).

[NA08] T. T. Nguyen and G. Armitage. “A survey of techniques for Internet traffic classification using machine learning.” In: *IEEE Communications Surveys & Tutorials* (2008).

[Shi+16] W. Shi et al. “Edge computing: Vision and challenges.” In: *IEEE Internet of Things Journal* (2016).

[Sna+91] S. R. Snapp et al. “DIDS (Distributed Intrusion Detection System) - motivation, architecture, and an early prototype.” In: *National Computer Security Conference*. 1991.

[Som+17] G. Somani et al. “DDoS mitigation techniques in cloud computing: Challenges and opportunities.” In: *IEEE Communications Surveys & Tutorials* (2017).

[Tro19] M. Trotter. “Network traffic analysis in high-speed environments.” In: *International Journal of Computer Science and Network Security* (2019).

[VB17] P. Voigt and A. von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. 2017.

[Xu+11] K. Xu et al. “Internet traffic behavior profiling for network security monitoring.” In: *IEEE/ACM Transactions on Networking* (2011).