Analysis and Classification of Cyberattack Traffic using the SecGrid Platform

Jan von der Assen, Muriel Franco, Bruno Rodrigues, Burkhard Stiller
Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland
E-mail: [vonderassen, franco, rodrigues, stiller]@ifi.uzh.ch

***Abstract*—Distributed Denial-of-Service (DDoS) attacks remain a notorious threat to businesses and governments. As defense mechanisms and investments therein were extended, the dynamics of attacks have adapted accordingly. Not only have attacks gained in frequency and size, but the underlying attack vectors have also evolved. Thus, there is a need for capable tooling that allows researchers, operators, and decision-makers to obtain insights into the behavioral aspects of attacks and their impacts. Such tooling needs to be able to keep up with the dynamic nature and with strong scalability requirements. In this demonstration, the SecGrid platform is introduced to facilitate traffic analysis and visualization of volumetric data. Using the SecGrid's engine, a range of applications from behavioral visualization, impact estimation, or ML-based attack classification are enabled.**

***Index Terms*—Cybersecurity, Network Traffic Analysis, Cyberattacks Identification, Information Visualization**

I. Introduction

Forecasts on the state of Distributed Denial-of-Service (DDoS) attacks in 2021 have proven to be true, as mitigation service providers continue to reveal a highly active threat landscape. The danger presented by attacks that are evolving in scale, frequency, and complexity has led to increased adoption of mitigation tools and services [1].

To mitigate such attacks, various types of users need to be able to understand the characteristics at hand [2]. For example, while business decision-makers seek to understand the abstract implications of an attack, researchers wish to obtain a low-level view on an attack to discover patterns of an attack vector.

With that background, captured network traffic is a promising data source that can be used to analyze attacks. Besides capturing information in the form of logs directly from a host or application, flow capturing and packet capturing are prominent approaches [3].

While the flow-based data structure makes the approach preferable for the operation of high-speed networks, it lacks important data to be analyzed. Packet captures, on the other hand, provide full information on the traffic exchanged [4].

In order to gain insights from captured traffic, a plethora of software from industry and academia exists. However, many of these implementations are either not scalable to analyze volumetric attack traffic (*e.g.*, WireShark and tshark) or they are highly complex (*e.g.*, Hadoop and ELK stack) and require substantial infrastructure investments. All of these approaches do not expose simple to use visualizations targeting cyberattacks. The mapping process from packet data to useful insight is often left to the user [5].

The SecGrid platform contributes to addressing this issue by implementing an extensible approach to analyze network traffic with appropriate scalability and usability. Also, this enables a collaborative setting to share insights from volumetric network traffic. Based on that, insights can be created with a flexible implementation that allows additional approaches to be built on the top of the SecGrid, such as ML-based classifiers and useful visualizations to understand cyberattacks behaviors.

II. Overview

The SecGrid platform implements an automated traffic analysis process that is scalable to enable large-scale analysis. Since information sharing in a post-mortem setting is a key use case, the usability requirement for a number of stakeholders must be emphasized.

To enable users to create insights using any dimension of network traffic, the PCAP file format was used as primary input data. Providing a scalable solution on top of commodity hardware (*i.e.*, without the usage of specialized hardware or cloud environments) dictates a strict flow of data. Therefore, packets are streamed through the analysis engine, where they are inspected without collecting them first. After a stream of packets was reduced to an interim result, it is visually transformed. This stream-processing-like approach ensures that the analysis is not bound to a certain magnitude of input data size. Further, as shown in the discussed architecture, the components in this process are designed to support novel analysis techniques in a modular way.

A. Architecture

The architecture designed to implement the previously described process ensures the usability and scalability requirements while allowing the platform's extensibility in terms of data, visualizations, and features.

The architecture comprises three major components: *(i)* The User Layer contains the visualization subsystem. This component is invoked after the analysis has concluded and insights were created. This is the only interface exposed to the user through which he/she may submit a new analysis request. Doing so invokes the analysis process *(ii)*, which involves

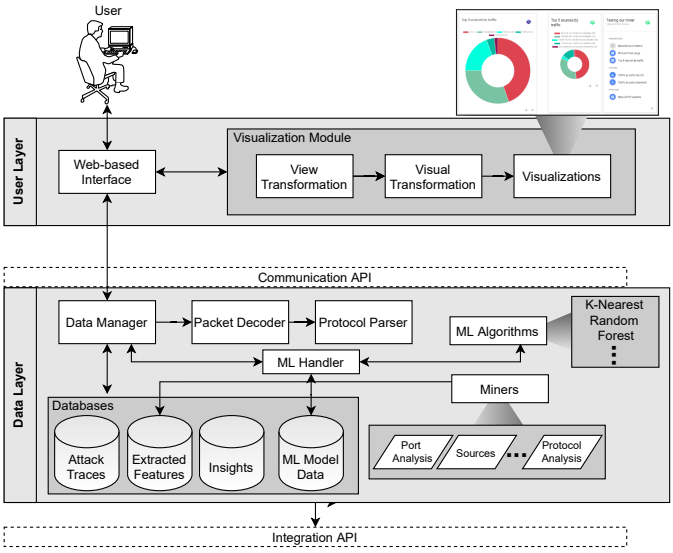

Fig. 1. High-level Architecture of the *SecGrid*Platform

decoding and analyzing the packets to extract insights. The insights are extracted by a set of miners. Designing the architecture this way allows that only one packet is kept in memory at any point in time. Finally, each miner produces the aforementioned insights that are then stored using a set of *(iii)* databases which mark the architecture's boundary. However, it is important to mention that the design allows the miners to enrich the extracted information using external databases [5].

B. Prototype Implementation

To present a working prototype, the architectural design described by [5] was implemented leveraging the *Node.js* platform. The visualization module of the User Layer is developed as a single Progressive Web Application (PWA) web technology.

The actual network analysis is implemented using pure JavaScript and a binding to the *libpcap* C++ library. The protocol parser is the core of the analysis module. It exposes an easy-to-use interface, where clients express that they wish to be called upon with a packet according to a specific configuration. Said configuration is expressed for a specific protocol (*e.g.*, UDP or TCP) or an abstract OSI-layer (*e.g.*, Transport-layer or Application-layer). This is done by the clients during the initialization phase where they load additional data sources to enrich the extracted features (*e.g.*, WHOIS databases or BGP Looking Glasses).

The clients, referenced as miners, process each packet individually, by reducing it to a dimension that embodies the insight. Internally, miners use numerous approaches to create insights during the extraction phase. For example, a simple miner may collect statistics on the protocols being used. Another miner independently extracts features (*e.g.*, the control flags of a TCP segment or their arrival timestamps) for a supervised learning algorithm. Random Forest (RF) and K-Nearest Neighbors (k-NN) algorithms were integrated into the post-processing phase of the prototype.

TABLE I
EXAMPLES OF THE MINERS IMPLEMENTED BY *SecGrid* [5]

Miner	Target Data	Outcome
Metrics Analyzer	Attack duration, number of packets, IPs and ports	High-level metrics used to fingerprint the attack
IEEE 802.1Q Tagging	Frame tags	Overview over the VLAN membership of link-layer frames
ICMP messages	ICMP headers	Overview over ICMP message types
IP Protocol Analyzer	IPv4, IPv6 packets	Analysis of the packets according to the IP protocol versions being used
Port Analyzer	UDP and TCP ports	Overview of the most used UDP/TCP ports by number of segments
Top Source Hosts Extractor	Source address	Overview of the hosts sending more traffic and requests
TCP States Analyzer	TCP flags	Analysis of the frequency of TCP flags in the packets, such as ACK, SYN, and FIN
Browser and OS Analyzer	HTTP User Agent	Identifies the browser and operation system being used for the request
HTTP Analyzer	HTTP Verbs and End-points	Analysis the most used HTTP requests (*i.e.*, GET and POST) as well as the end-points accessed via HTTP protocol
BGP Analyzer	BGP Messages	Messages exchanged between BGP speakers over time
ML-Feature	Events emitted by the Protocol Parser	Listens to all events emitted by the protocol parser and process the information required for the attack classification ML model

After the decoding phase, each miner finalizes the insight creation. For example, the ML-based miner either updates the model or classifies the attack. To validate the architecture, a set of miners were implemented to extract and enrich insights, as shown in Table I.

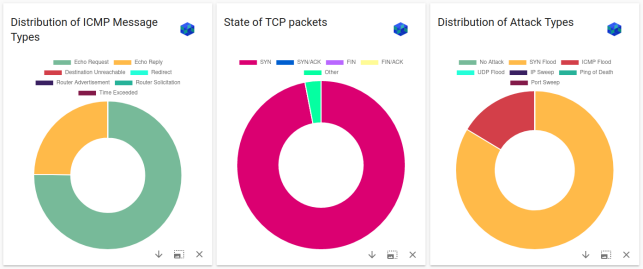

Fig. 2. Automated Classification (right) and Manual Classification (left, center) of a Multi-vector DDoS attack

III. SETUP AND DEMONSTRATION

The demonstrator presents how researchers use the ML-based classifiers of the analysis platform in a collaborative setting. In that, there are two instances of SecGrid deployed on distinct servers. An instance of the attack-sharing platform DDoSDB is used as an intermediary to exchange insights. All three services are deployed using virtual. Each machine is assigned one CPU core and 1 GB of memory. Access to and between the services is provided through a reverse proxy.

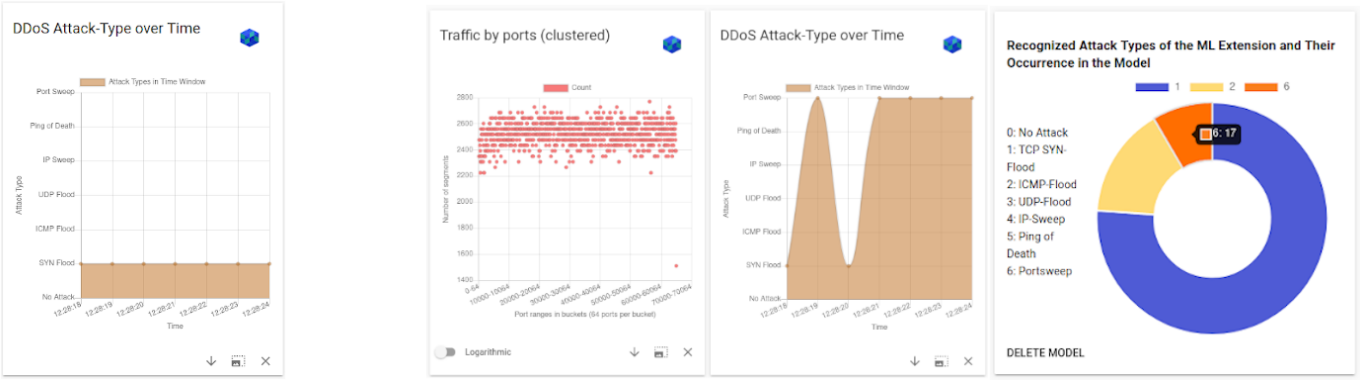

Fig. 3. Falsely Classified Attack Trace (left) and Reclassified Trace based on Refined Model (center, right)

Two datasets were created using the CLI tools *tcpdump* and *nmap*. Both datasets hold traffic of a TCP-SYN-based port scan. Before the demo, dataset *A* is analyzed in the *SecGrid A* instance and published to *DDoSDB*. *SecGrid B* holds the unclassified dataset *B* and a machine-learning model *M* trained on TCP-SYN-flooding and ICMP-based attacks. Figure 4 depicts the configuration of datasets and services used for the demonstration.

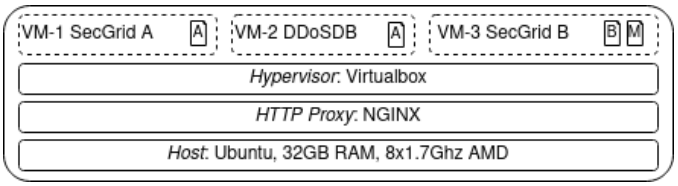

Fig. 4. Overview on the services and datasets deployed at the start of the demonstration

The demonstration of the process shown in Figure 5 starts with the second researcher automatically classifying the attack trace *B*. The model was previously trained based on network traces labeled as SYN-flooding attacks. Therefore, the attack is falsely classified due to the similarity of the attack vectors. This is contrasted by manually interpreting the visualizations that highlight the number of segments received per destination port.

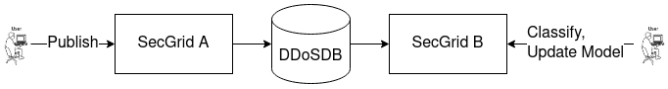

Fig. 5. Collaborative process between two researchers exchanging labeled attack data

The presenter then characterizes the machine learning model *M* using the information stored in the SecGrid system, revealing that the model is biased towards the SYN-flooding-based attack vector. Thus, he/she navigates to the DDoSDB platform to obtain shared attack traces of other attack vectors. There, the demonstrator highlights how the user retrieves insights previously created in the *SecGrid A* instance.

Finding the appropriate attack is supported by the query language in DDoSDB and comments submitted by other researchers. These traces *A* are then imported into the *SecGrid B* instance. During that, the imported attack is manually classified for the supervised learning algorithm. After retraining the local model, the existing trace *B* is reclassified. Although that trace is significantly larger, it is classified instantly. Finally, it is presented how the accuracy of the classification was refined using the collaborative features of the system by visualizing attack trace *B*, as shown in Figure 3.

IV. Summary and Conclusions

This demonstration shows how the SecGrid platform can be applied to a collaboration process of researchers exchanging attack traces to update an ML-based model. The presented elements highlight the necessity of easy-to-use and efficient feature extraction tools to allow collaborative work to be performed on that. Due to the modular architecture proposed by the SecGrid approach, future research is carried out on various levels of the architecture. Such approaches include the analysis of traffic in real-time and data analysis techniques such as federated learning.

Acknowledgments

This paper was supported partially by *(a)* the University of Zürich UZH, Switzerland and *(b)* the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 830927, the CONCORDIA project.

References

[1] Akamai, "2021: Volumetric DDoS Attacks Rising Fast," March 2021, https://blogs.akamai.com/2021/03/in-our-2020-ddos-retrospective.

[2] G. Gallopeni, B. Rodrigues, M. Franco, and B. Stiller, "A Practical Analysis on Mirai Botnet Traffic," in *IFIP Networking Conference (Networking 2020)*, Paris, France, 2020, pp. 667–668.

[3] D. Freet and R. Agrawal, "A Statistical Comparison of Security Visualization Efficiency Compared to Manual Analysis of IDS Log Data," in *SoutheastCon 2018*, 2018, pp. 1–5.

[4] G. Harris, "PCAP Capture File Format," December 2020, https://tools.ietf.org/id/draft-gharris-opsawg-pcap-00.html.

[5] M. Franco, J. von der Assen, L. Boillat, C. Killer, B. Rodrigues, E. J. Scheid, L. Granville, and B. Stiller, "SecGrid: A Visual System for the Analysis and ML-Based Classification of Cyberattack Traffic," in *IEEE 46th Conference on Local Computer Networks (LCN 2021)*. Edmonton, Canada,: LCN, October 2021, pp. 1–8.