2021 IEEE 46th Conference on Local Computer Networks (LCN) | 978-1-6654-1886-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/LCN52139.2021.9524932

SecGrid: a Visual System for the Analysis and ML-based Classification of Cyberattack Traffic

Muriel Franco[1], Jan Von der Assen[1], Luc Boillat[1], Christian Killer[1],
Bruno Rodrigues[1], Eder J. Scheid[1], Lisandro Granville[2], Burkhard Stiller[1]
[1]*Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH*
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland
[2]*Computer Networks Group, Institute of Informatics, Federal University of Rio Grande do Sul UFRGS*
Av. Bento Gonçalves, 9500, Porto Alegre, Brazil
E-mail: [franco, vonderassen, killer, rodrigues, scheid, stiller]@ifi.uzh.ch[1],
lucthierrynicolas.boillat@uzh.ch[1], granville@inf.ufrgs.br[2]

Abstract—**Due to the increasing number of cyberattacks and respective predictions for the upcoming years with even larger numbers of occurrences, companies are becoming aware not only that the digitization of their businesses is essential, but also that the adoption of efficient cybersecurity strategies is crucial. Therefore, approaches for a better understanding and analysis of cybersecurity are essential.**

Thus, SecGrid, a Machine Learning (ML) empowered platform for analyzing, classification, and visualization of cyberattacks is introduced. SecGrid implements an extensible set of miners to analyze information from network traces to provide insightful visualizations of malicious traffic given and to classify automatically different types of cyberattacks by using supervised ML. Experiments conducted show high overall usability, scalability in terms of the capacity of the platform to extract information from large files, and high performance and accuracy during the classification of cyberattacks.

I. Introduction

The number of cyberattacks continues to rise in many sectors due to business competition and cyberwarfare [24]. Most of these attacks negatively impact companies or governments [7], such as ransomware being used for extortion, Distributed Denial-of-Service (DDoS) attacks against business competitors, and phishing as a door that leads to data leakage. Analytical approaches are required to understand cyberattacks better and conduct planning tasks required to handle them. These approaches include tools for analyzing network traffic, extracting information for pattern recognition, and visualizing traffic behavior.

Network traffic analysis has been used for different purposes, *e.g.*, to monitor and execute tasks related to performance, accountability, and security. For that, popular tools [22] capture and analyze network traces (*e.g.*, Packet Capture (PCAP) files and Netflow records). In the cybersecurity field, a postmortem analysis (*i.e.*, after the attack occurred) read traffic log files in order to extract characteristics of the attack, identify damages, and acquire sufficient knowledge to mitigate new attacks [12]. However, there still exists a certain lack of visualization approaches to explain cyberattacks and to support cybersecurity planning.

Although there are well-known and commercially established tools for the analysis of security-relevant data placed in the market (*e.g.*, Elastic Stack (ELK) and Splunk) [3], which are popular among Chief Information Security Officers (CISO), there are open challenges with respect to the security analysis and opportunities for advanced tools that simplify the understanding of cybersecurity with a limited need of dedicated staff and cybersecurity skills. Such tools are not trivial since providing security reports based on log files is challenging due to varying information and entropy, such as the semantic mapping of header fields, correlation of information, and specific-protocols characteristics.

Besides the support of different tasks (*e.g.*, forensics, education, and proactive measures), such a tool can be used as an ally for data preparation required for the classification of cyberattacks empowered by Machine Learning (ML) techniques [18]. Also, ML-based systems can benefit from insightful visualizations to represent data in an accessible manner for different stakeholders without prior knowledge within such a field. Thus, these approaches can be used during the process of taking action against further attacks, such as those related to DDoS attacks and malware infection.

Thus, this paper introduces *SecGrid*, an open-source platform for postmortem analysis, classification, and visualization of cyberattacks [5]. *SecGrid* addresses the lack of integrated approaches for processing, analyzing, and visualizing complex datasets of cyberattacks by implementing an extensible set of miners to process information from network traces (*i.e.*, PCAP) and providing visualizations for the analysis of cyberattacks. According to demands, both miners and visualizations are extensible to address different scenarios and requirements. *SecGrid* is based on an ML-based approach to automatically classify traffic given according to its type (*e.g.*, SYN flood and Ping of Death) or as regular traffic. Experiments were conducted to evaluate *(a) SecGrid*'s overall usability, *(b)* the miners' performance and scalability to process information from different datasets composed of real-world cyberattacks, and *(c)* the accuracy of the ML-based attack classification.

The remainder of this paper is organized as follows. Section II reviews related work. While Section III introduces *SecGrid*, Section IV contains the evaluation followed by conclusions and future work as of Section V.

978-1-6654-1886-7/21/$31.00 ©2021 IEEE

II. Related Work

Understanding a cyberattack and its impacts involves the analysis of large and complex datasets to observe characteristics and relationships. Visualization techniques (both live and post-mortem) were applied for these purposes along the years [6], because a visual representation of a dataset provides insights for a human observer. For example, AfterGlow is a security visualization tool [17] that facilitates the creation of graphs (*e.g.*, linked and network graphs) to visualize datasets to understand relationships between different entries. Furthermore, Wireshark can capture packets and provides a set of plots to analyze better the traffic captured (*e.g.*, plotting of TCP flags and network throughput) and to conduct, for example, intrusion detection [22]. Currently, the most prominent approach is ”Elastic Stack”, composed of the integration of Elastic Search, Logstash, and Kibana [3].

However, most of the current tools are complex (*e.g.*, higher level of granularity, hard for non-experts to conduct an analysis, and many interactions are required to deploy and obtain insights) to use or too generic in terms of their visualizations (*i.e.*, not focusing specifically on cybersecurity, but on more generic monitoring tasks). Thus, their use is limited only to network experts and professional penetration testers with particular technical expertise. Besides these factors, additional limitations are observed in terms of scalability, since some tools (*e.g.*, Rumint and Wireshark) do not scale with (very) large PCAP files [17], which becomes a problem when considering the ever-increasing load of cyberattacks. This negatively impacts the adoption of cybersecurity monitoring by companies and delays the definition of efficient cybersecurity strategies, which requires sufficient information about threats with which businesses are being targeted.

Therefore, efforts are required from both industry and academia to simplify the process of understanding cyberattacks, while empowering decision-makers to adopt better cybersecurity strategies based on observable events [8]. Today that becomes even more challenging, given that, for example, 5G becomes a reality together with a highly connected world, *i.e.*, with more vulnerable devices, complex networks, and financial incentives for cyberattacks [19].

Besides approaches that support the analysis and visualization of security aspects, the combination of ML techniques and cybersecurity have been applied to the classification of attacks. A comparative study of different ML techniques and their performance is introduced in [21]. [15] achieved an accuracy of 98.6% with a Random Forest classification for DDoS attacks; however, no other metrics were presented, except for the false-positive rate of 2.4%. They used a combination of flow-based and pattern-based features for their classifiers. [14] achieved a Random Forest classification accuracy of 99.53%, a precision of 100%, a recall of 99.12%, and an F1 score of 0.9956 in a joint detection solution, which involved multiple hosts analyzing traffic. Also, [20] achieved an accuracy of 96.6% and an F-Score of 0.969 using the k-Nearest Neighbor classification. This small investigation of the state-of-the-art of the ML field for cyberattacks classification indicates a path to follow since opportunities for novel approaches (*e.g.*, considering different features, datasets, and techniques) and improvements of existing ones are exploited.

III. The *SecGrid* Platform

Three main dimensions are defined to represent the requirements of *SecGrid*: *(a)* Automation, *(b)* Usability, and *(c)* Scalability and Extensibility. A fully integrated and automated process is provided to process and store information from PCAP files containing traffic originating from cyberattacks. Different levels of abstractions are provided during the analysis and comparison of attacks, which simplifies identifying characteristics related to a cyberattack for improved cybersecurity planning and enhanced detection of attacks. Also, a supervised Machine Learning (ML) approach is used during the analysis and classification of cyberattacks. Finally, *SecGrid*’s components are designed to be extensible, which means that both the extracted information and the visualizations can be rendered according to user-specific demands.

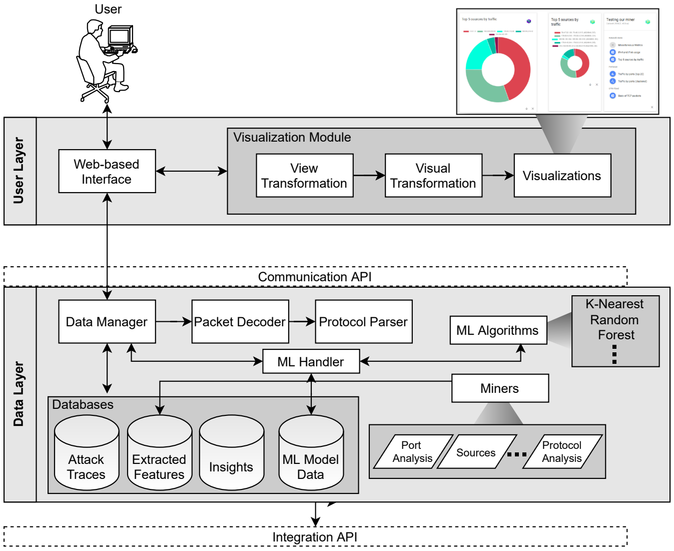


Fig. 1: *SecGrid*’s Architecture

SecGrid consists of *(i)* miners, which are able to decode PCAP files and extract features from different protocols (*e.g.*, Ethernet, IP, TCP, and HTTP), *(ii)* a web-based interface that allows users to interact with the platform and access overview statistics, *(iii)* a ML model to automatically classify different attacks traffic, and *(iv)* visualizations that give insights regarding the datasets under investigation. Besides, a web-based User Interface (UI) is provided, where users can share their insights and datasets with interested users, taking into account privacy concerns (*i.e.*, data anonymization).

As of the architecture of *SecGrid* (*cf.* Figure 1) the user accesses the Web-based interface to analyze a dataset available (*i.e.*, PCAP file) or upload a new one. The *Data Manager* is in charge of handling user requests to store and access data related to a cyberattack. When the user uploads a new dataset, it is forwarded to the components of the *Data Layer*, which perform data extraction and processing of all relevant features of the cyberattack. Finally, these features are available in a well-defined data structure for the *Visualization Module* to build different visualizations according to user interactions. The *Data Layer* and the *User Layer* communicate through the *Communication API*. An *Integration API* allows for external solutions to request information and reuse available miners, which provides integration options, such as for systems to recommend or offer protections against cyberattacks [9], [10].

A. User Layer

The *Visualization Module* renders the diagrams based on the result produced by the *Data Layer modules*. This module contains three components that work together and can be implemented as one single integrated module. First, the *View Transformation* component receives the results of the mining process through the *Communication API*. It then transforms the properties of the data structures into fitting visualizations, such as by mapping the number of packets contained in a result to the values for a y-axis of a bar chart.

These data structure properties can then be further adapted by the *Visual Transformation*, which can change the way how these properties are shown. *e.g.*, given that a user wants to zoom out, it aggregates certain data points that would be on the y-axis of a bar chart. Finally, with this configuration, a set of visualization components can plot the data using different visualization techniques. For example, a set of TCP ports and their number of occurrences could be plotted using different types of charts (*e.g.*, line, bar, histograms, pie). Therefore, the *Visualization Module* contains the required components that allows *SecGrid* to built interactive visualizations.

Each visualization is described as a template, which can be feed with different information, thus allowing for reuse of the same features to visualize different behaviors (*e.g.*, a malware in its infection phase or a DDoS attack on the application layer). New visualization templates can be added to enrich the capacity of *SecGrid* to plot the information available in the databases.

Figure 2 provides an overview of the *SecGrid* Web-based Interface. At this view, the opened dataset (*i.e.*, Dataset 1) can be seen on the top left. In the *Metrics Tab*, a summary of all extracted information from this dataset is available (*e.g.*, number of packets, attack size, and the number of different sources IPs). All of the visualizations are accessible via the *Visualizations Tab*, separated based on the OSI layer model (*e.g.*, visualizations considering the application, transport, network, and physical layers). After clicking on one visualization, a new window is added to the dashboard grid, allowing for the analysis of different information and even datasets simultaneously. For example, these opened visualizations can indicate a possible SYN Flooding attack leading to port 80 with a specific origin (*e.g.*, IP addresses and country).

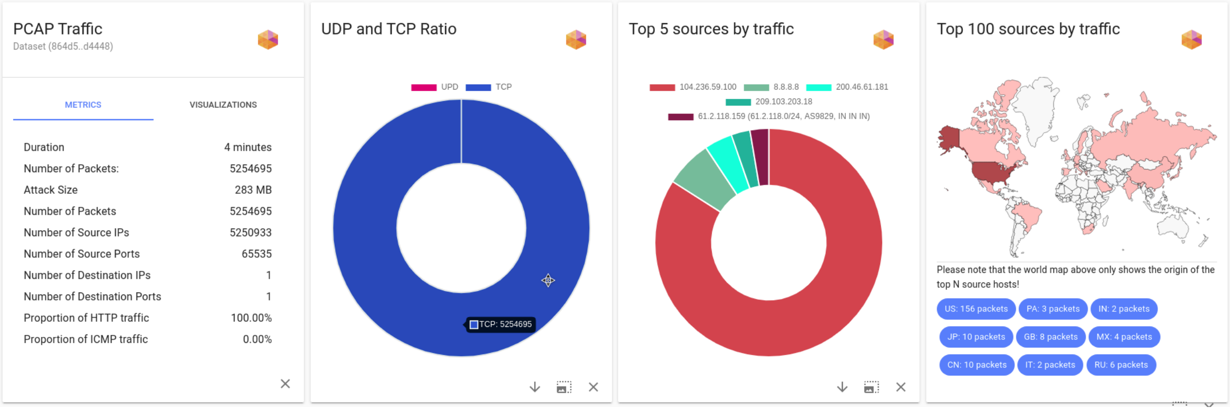
B. Data Layer

The *Packet Decoder* module reads a PCAP file provided by the *Data Manager* and parses packets using the *Protocol Parser*, which is in charge of identifying the type of packets and separate them for further analysis. Then, the *Miners* extract specific information from the decoded packets, thus making the extracted features available to users in different ways (*e.g.*, as a statistical report or in a set of insightful visualizations). These modules allow independent miners to observe some protocols' packets to analyze them without fetching packets that they do not need or without being affected by other miners. The protocols decoded and processed today by *SecGrid* include: Ethernet, ARP, IPv4 and v6, ICMP, TCP, UDP, and HTTP.

TABLE I: Examples of the Miners Implemented by *SecGrid*

Miner	Target Data	Outcome
Metrics Analyzer	Attack duration, number of packets, IPs and ports	Overview of metrics associated to a cyberattack log file
IEEE 802.1Q Tagging	Frame tags	Overview over the VLAN membership of link-layer frames
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
Device Analyzer	HTTP User Agent	Identifies which type of device is being used for the request
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
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

This set of miners (*cf.* Table I) can access packets of a particular protocol or abstraction. Thus, it allows for independent feature extractors that focus on producing a visualization result for one or more attack types. For example, one implements an autonomous miner to visualize a possible SYN flood attack and an additional miner for a DNS amplification attack. The first miner observes packets being emitted in TCP protocol, and the latter observes an application layer protocol. Then, they extract relevant features from the packets and finally store them for analysis. For example, a TCP States miner analyzes the TCP flags that indicate the connection states, highlighting the distribution of connection states of the overall observed packets. The miners then make available the results in a structured way for the visualizations of such characteristics. It is important to note that although a set of miners are provided, it can be extended by implementing new miners (*e.g.*, JavaScript or Python-based) and linking them with the *Packet Decoder*. Thus, *SecGrid* extensibility allows for further



PCAP Traffic

Dataset (864d5..d4448)

METRICS	VISUALIZATIONS
Duration	4 minutes
Number of Packets:	5254695
Attack Size	283 MB
Number of Packets	5254695
Number of Source IPs	5250933
Number of Source Ports	65535
Number of Destination IPs	1
Number of Destination Ports	1
Proportion of HTTP traffic	100.00%
Proportion of ICMP traffic	0.00%

UDP and TCP Ratio

Top 5 sources by traffic

Top 100 sources by traffic

Please note that the world map above only shows the origin of the top N source hosts!

Fig. 2: *SecGrid*'s Graphical Overview (Example)

analysis and insights about different types of cyberattacks and behaviors, such as traffic related to Wannacry (Ransomware), Mirai (Botnet), or even a simple port scanning.

Table I lists examples of miners implemented by *SecGrid*. All these miners can be applied separately or combined to extract meaningful information about the traffic available in a PCAP file. This information can then be combined to generate different reports and visualizations. For example, the combination of the TCP States Analyzer, Device Analyzer, and Port Analyzer can be used to identify Command and Control (C/C) traffic between IoT devices of a botnet. This traffic usually shows activity in port 23 (*i.e.*, Telnet), with most of the packets being SYN packets [11] and occasionally keep alive packets (*e.g.*, PSH and ACK) can be observed [1].

For the classification of given attack traffic, the *ML Handler* acts as a gateway for the *SecGrid* and the ML algorithms (*e.g.*, K-Nearest Neighbour, Random Forest, or Neural Networks). Thus, when traffic data has to be classified, the *Data Manager* sends a request to the *ML Handler*, which will be in charge of preparing the data to be used as input for the implementation classification ML algorithms. Also, the *ML Handler* manages the *ML Model Data*, training and adding new data from the ML-Feature miner when requested. Details on the implementation and the ML models being used are presented in the following sections.

C. Prototype and Implementation

All of the *SecGrid*'s components and its *Web-based interface* were developed using Javascript (mostly Node.js and Vue.js). An instance of a running prototype and its source code is publicly available at [5]. Integration with the European DDoS Clearing House pilot is placed [8] to support better the analysis of one of the most prominent cyberattacks: DDoS attacks, thus allowing for the exchange of information and features between *SecGrid* and the DDoSDB. Thus, *SecGrid* and the DDoSDB can communicate via APIs provided by both solutions. The need for the development of new miners is because extensibility and control of the different levels of granularity possible are important when extracting information to provide insights, novel visualizations, and features for today's and next generation of cyberattacks.

One of the most important decisions when implementing the packet parser of *SecGrid* is which library to use for network capture decoding. The final decision for a library based on Node.js was taken because of different reasons, such as *(a)* technical aspects related to the capacity to handle capture files that are multiple gigabytes large and *(b)* scalability aspects, which require well-defined structures to allow all miners to be easily extendable and optimized. Besides that, to support the implementation of the different miners, a list of protocol parsers was initially developed for the *SecGrid*. Table II lists protocols that are decoded by platform's *Protocol Parser*. Besides those protocols listed, other protocols and packet types are already decoded and emitted to miners (*e.g.*, 802.11, IGMP, SSL, and Websocket). Thus, it is possible to implement miners to extract features from them according to the demands of the different scenarios.

TABLE II: Decoded Protocols by *SecGrid*'s Protocol Parser

Protocol	Description	Decoding support
Ethernet	This network access layer protocol was observed most frequently during the initial investigations	Fully decoded packets can be obtained and the properties, including 802.1Q tags, can be conveniently accessed
ARP	The address resolution protocol provides resolution services to the internet layer	Decoding fully supported
IPv4	Version four of The internet-layer IP protocol	Full access to all properties in the IPv4 header
IPv6	Version six of the Internet-layer IP protocol	Limited decoding support. Only the fixed frame header is decoded with limited support for extension headers
ICMP (v4)	The control protocol used along IPv4	Decoding fully supported
TCP	Widely used connection-oriented transport-level protocol	Decoding fully supported
UDP	Widely used connectionless transport-level protocol	Decoding fully supported
HTTP	Application-level protocol	Parsers for certain attributes have been implemented, e.g. User-Agent strings
BGP	Exchange routing protocol used by autonomous systems on Internet	Parsers for BGP messages

Multiple miners implemented by *SecGrid* provide specific features for the analysis and understating of cyberattack traffic. Also, they allow for an easy extension of metrics and scenarios to be considered, which is useful for different purposes, such as research experiments, cybersecurity education, and companies with specific analysis demands. However, as an alternative for *SecGrid* miners, well-known network tools like *tcpdump* and *SmartSniff* can use *SecGrid* as a more intuitive visualization and reporting platform. Thus, these powerful tools can be integrated into the *SecGrid* dashboard, also enabling opportunities for *SecGrid* real-time analysis by using tools already tested and validated by the cybersecurity market.

D. ML-based Attack Classification Model

Although miners as implemented (*cf.* Table I provide information for the analysis of attacks, it is still required to process this data in order to be used to build the ML training model. Therefore, a miner *ML-Feature* was designed to extract data available by other miners and transform them to build the ML training model, such as transforming respective source IP addresses and ports into unique counters, obtaining the percentages of packet types from packets extracted, and calculating the inter-packet interval from the array containing all arrival times of packets.

The special *ML-Feature* miner listens to all possible events emitted by the package parser and processes all information of a time window to provide all relevant features for the attack classification. Thus, the *ML-Feature* miner is used to process the PCAP files in order to generate the required information to be used in the training phase of *SecGrid*'s algorithms and also during the classification of attacks. The training dataset created and to be used by *SecGrid* consists of 55,349 records extracted from different DDoS attack datasets, publicly available at [5]. Thus, as a proof-of-concept, the implementation and training dataset is provided to classify seven different behaviors within *SecGrid*: Regular traffic, SYN Flood, ICMP Flood, UDP Flood, IP Sweep, Ping-of-Death, and Port Sweep.

Two ML algorithms were implemented in the *SecGrid* platform: Random Forest (RF) and k-Nearest Neighbor (k-NN). The Scikit-learn classifiers [23] were used for these algorithm implementations, allowing for customization options and parameter tweaks. The RF classifier used ten estimators, thus, ten sets of random decision trees were created and compared to each other. An evaluation was conducted in order to provide evidence on their accuracy (*cf.* Section IV-C).

After the classification of these different attacks identified in a single or multiple PCAP files, *SecGrid* provides a visualization in which the user observes along time, when regular traffic or specific attack occurred. This visualization achieves a clear view of the duration and behavior of attacks identified. Figure 3 shows on the left the total distribution of the attack classified in the given time. The darker the lines, the more intense the classified traffic is. Therefore, solid black lines highlight intensive traffic (in terms of packets per second), while gray lines show less expressive traffic. In addition, on the right side, a pie chart summarizes the percentage of packets representing each attack classified.

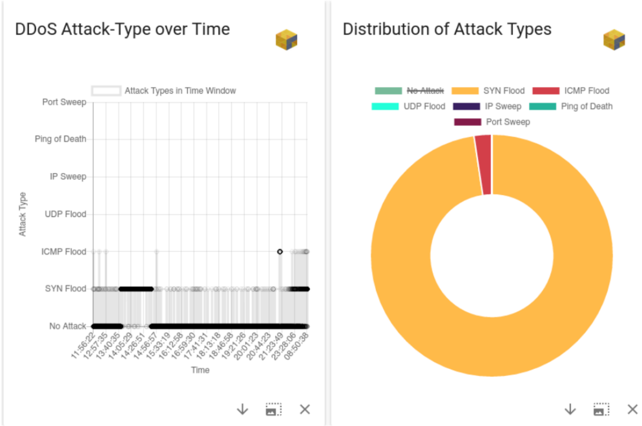


Fig. 3: Visualization of ML-based Classification of DDoS Attacks Along Time

IV. Evaluation

The evaluation of *SecGrid* addressed usability, performance, scalability, and the ML classification accuracy.

A. Survey and Usability

An online survey was conducted to analyze the usability of *SecGrid*. This was based on eight tasks selected to be performed and followed by a System Usability Scale (SUS) questionnaire [4]. The survey was conducted anonymously with 23 participants from different countries and institutions (both industry and academia), with different levels of scholarship (Bachelors, Masters, and Doctors), and with expertise in Computer Science-related areas (Computer Networks, Information Systems, Software Engineering, and Cybersecurity). The participants' age varied from 23 to 60 years, all of them knowing at least three or more types of cyberattacks. After filling in the initial information regarding their fields, scholarship, and previous knowledge, each participant was requested to watch a three-minute video introducing the main features of *SecGrid*. The participants were required to answer the questions as of Table III using *SecGrid*, resulting in respective success rate (*i.e.*, how many users answered correctly).

Most of the tasks achieve very high success rates. However, T5 and T7, even with binary answers (Yes/No), provided results slightly above 60%. This is because some of the participants did not use the visualizations provided to answer the questions but relied mainly upon the overview of statistics provided by the platform. This resulted in wrong answers since an SYN Flooding (the correct answer for T5) can be easily confused with an HTTP flood (most of the wrong answers for T5) without a more in-depth check. The same happens for T7, in which some participants did not analyze the whole traffic but only the most used port (HTTP 80). Therefore, it is essential to organize both visualizations and statistics to improve usability and avoid misinterpretation, thus guiding the users to the most accurate insight possible.

Besides these tasks conducted and according to the answers of the SUS questionnaire, the majority of users found

SecGrid easy to use (91.3% of the answers) and well-integrated (91.3%). Also, users were confident to use the system (82.6%) and would like to use *SecGrid* frequently (78.3%). Although most participants' feedback was positive, suggestions to improve the usability were provided at the end of the survey. The suggestion that appeared most frequently refers to the feature to freely create and save a dashboard using available visualizations and datasets. Thus, this featured was implemented additionally to improve the overall usability. However, other relevant features mentioned by participants can improve the users' capacity to gain insights using *SecGrid*, such as those related to the analysis of datasets based on time-series plots, more contextual information, and support for insights sharing.

TABLE III: Tasks Performed by Users using the *SecGrid*

Task	Answer	Success Rate
T1: How many packets are present in the dataset?	1640892	95.7%
T2: How many hosts participated in the attack?	2678	91.3%
T3: From which part of the world were most packets sent?	Asia	91.3%
T4: Which destination port received the largest number of segments?	80	95.7%
T5: Was the traffic only sent to ports in the 'well-known port range' (1-1024)?	No	69.6%
T6: Which of the following attack vectors describes the attack in dataset "Dataset 1" best?	SYN Flooding	60.9%
T7: Regarding the HTTP traffic in the "Dataset 2", would you consider this traffic as being part of the attack?	No	78.3%
T8: Looking at the metrics, which of the following attack vectors describes the attack in "Dataset 2" best?	ICMP Flooding	69.6%

B. Miners' Performance and Scalability

In order to investigate the performance of the *SecGrid*'s miners and the scalability of the platform, the time to extract the features from different PCAP files was measured, as well as the RAM memory and CPU used during the process of open, extracting, and store information from these files. The experiments were conducted with *SecGrid* instantiated in a container-based application limited to the usage of 2 GB of RAM running in an Intel Core i7-8650U, with a base frequency of 1.9 GHz, 16 GB of RAM with a clock speed of 2.133 GHz. All of the scripts used to generate the results are available at [5].

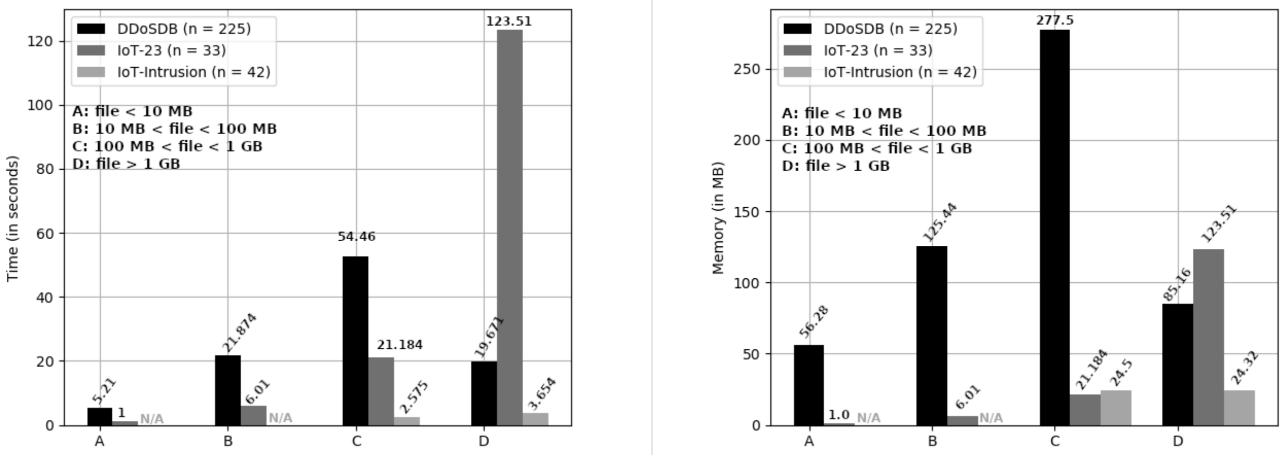
For the analysis, we are considering three different datasets of PCAP files: *(a)* the DDoSDB [16], which provides data from collaborators that usually collected the data as a victim, *(b)* the IoT-23 [2], a dataset of network traffic from IoT devices captured by the Stratosphere Laboratory of Czech Technical University in Prague during the years of 2018-2019, and *(c)* the IoT Network Intrusion dataset [13], which contains various types of network attacks simulated in IoT environments for academic purposes. The process was conducted as follows. First, the dataset files were stored locally. Then, for each one of the dataset log file, the miners of *SecGrid* were applied to extract information from that, and the time and memory used to process each file was recorded. Also, it was recorded if the datasets were successfully processed or not. Finally, besides processing each file, the average time to process each dataset was calculated. This approach was applied in 10 different rounds to have more statistically accurate results. Thus, 300 PCAP files were tested in total, with a total size of 112 GB in datasets. The number of files in each dataset is identified in the experiments using the variable `n`.

Figure 4 presents the overview of the tests performed with the *SecGrid*'s miners. For a better analysis of the results, the three datasets were separated into four different scenarios (*i.e.*, A, B, C, and D) based on their files' size. Figure 4 (a) shows the time to process each one of the datasets. It is possible to see that the response time for smaller files (*i.e.*, those presented in the IoT-Intrusion dataset) seems to increase linearly. Still, when comparing other datasets (*e.g.*, DDoSDB), it is possible to observe that the time required to process files between 100 MB and 1 GB (Scenario C) is higher than to process the files higher than 1 GB (Scenario D). This happens because the response time is dependant on the size of the file and the entropy present on the file (*e.g.*, how much information, complexities of the packets, and characteristics have to be extracted). The dataset that showed large files with higher entropy was the IoT-23 [2]. The average time to analyze the files with more than 1 GB (Scenario D) in the dataset IoT-23 with high entropy was roughly 2 minutes. However, in this dataset, the worst scenario observable required roughly 25 minutes (1482.125 seconds) for a file, representing a Gafgyt Malware traffic, with 22.1 GB of size and very high entropy. It was not the larger file analyzed but the one that required more time to be processed.

The RAM memory is another critical resource for the *SecGrid* scalability and stability. Therefore, its usage by miners was also measured during the experiments. Figure 4 (b) provides the average memory used to analyze each one of the PCAP files. All 300 PCAP files were successfully analyzed, with the highest recorded memory consumption being around 1 GB for a file of 50 GB of a real-world DDoS attack. On average, the highest consumption was 277.5 MB for the dataset of DDoSDB (Scenario C), representing a log file of massive TCP Flood. Based on the tests performed, it is possible to ensure that *SecGrid* can process files up to 50 GB without any restriction. It was also measured that the Input/Output medium (*e.g.*, Hard Disk Drive or Solid-state Drive), from which the PCAP file is read, has to provide read speeds of at least 50 MB/s; otherwise, it might become a bottleneck for files with huge sizes and high entropy.

C. ML-based Classification Evaluation

For the evaluation of the performance and feasibility of the ML-based classification implemented inside of *SecGrid*, different algorithms and datasets were tested, resulting in an overall accuracy of 99.9% for both Random Forest (RF) and k-Nearest Neighbors (k-NN) implemented algorithms. The final model has a total length of 55,349 records and contained 18



(a) Average Time to Analyze PCAP Files

(b) Average Memory Consumption to Analyze PCAP Files

Fig. 4: Performance Evaluation of *SecGrid* for 300 PCAP Files

datasets, with most of them containing two to four attacks. The defined model had a size of 8.9 MB, which was only a fraction of the original datasets that were often gigabytes in size. For this, a model is trained using data from different data sources that include various DDoS attack types. This also includes regular traffic so that the system does not accidentally classify regular network states as attacks. The model was cross-validated using an 80% train and 20% test split. This cross-validation was performed ten times with a randomized dataset to split. The reported results in form of *Precision*, *Recall*, *F1-Score*, and *Accuracy*. The evaluation procedure was run with two different setups: without duplicates in the dataset (*i.e.*, removing all repeated occurrences of a record in the model) and with duplicates kept.

The results of each evaluation setup are displayed as macro averages (overall classification of all attacks without distinction of the records available in the model) and as weighted averages (calculates the metrics for high and low-occurrence of records for each attack type in the model separately). This helps to shows that not balanced occurrences of records can result in a misrepresentation of attacks in the dataset, which is masked by the correct classification of other attacks. For example, in a model comprised of 95% of TCP flood records and 5% of UDP records, even if the algorithm performs very poorly for UDP flood classification, the total metrics would still result in a good classification supported by many TCP attacks correctly classified. The accuracy was calculated by rounding a floating-point value, while *Precision*, *Recall* and the *F1-Score* were calculated by averaging percentage integers.

TABLE IV: Evaluation of Random Forest (RF) and K-Nearest Neighbors (k-NN) After Cross-validating the Built Model

Method	**Precision**	**Recall**	**F1-Score**
RF with Duplicates	100%	100%	100%
RF with Duplicates (Weighted)	95.1%	92.3%	92.4%
RF without Duplicates	100%	100%	100%
RF without Duplicates (Weighted)	97.9%	95.9%	96.5%
k-NN with Duplicates	100%	100%	100%
k-NN with Duplicates (Weighted)	83.7%	81.2%	82.3%
k-NN without Duplicates	100%	100%	100%
k-NN without Duplicates (Weighted)	85.4%	83.5%	84.3%

Table IV summarizes the results of the tests. The Precision, Recall, and F1-Score in all unweighted tests reached 100%, from which it is possible to infer that the system performs very well with attack-types for which many records exist in the database. Most records consisted of regular traffic, SYN Flood Attacks, ICMP Flood Attacks, and UDP Flood attacks in this test. Even though not all attack states were classified correctly, the large ratio of attack types classified 100% correctly to falsely classified types still averaged out to 100%. When comparing those values to their weighted counterparts, it becomes clear that the system had trouble classifying attacks that were not well represented in the dataset, such as IP Sweeps and Port Sweeps. Therefore, if the dataset had more records of these underrepresented attack types, the metrics could be closer to their unweighted counterparts.

All tests that had duplicate records removed performed better than their counterpart that kept the entire dataset. Duplicate records shrunk the model data to 43,714 records, which means a 21% reduction in length. The algorithm RF performed significantly better in the weighted result scores. This is because the RF classifier manages to classify low-occurrence attack types compared to k-NN classification better correctly. However, both algorithms performed equally well in classifying the well-represented attack types in the model.

Additionally, the time required to build the model with more than 50,000 records was 651 ms, with a classification time of 27 ms for the RF algorithm and 204 ms for the k-NN algorithm. This shows that, based on the time required for the classification, real-time analysis is possible, and RF can be a possible candidate for a production environment. When comparing these values achieved by *SecGrid*'s classification with other state-of-the-art approaches available in the literature, it is possible to verify the excellent results achieved by the *SecGrid*'s approach, especially from the RF classifier.

V. Conclusions and Future Work

This paper introduced *SecGrid*, an open-source approach for the analysis and visualization of cyberattacks. A running prototype and its source code are available at [5]. *SecGrid* stands as a planning and support tool to help network operators and decision-makers to gain insights about possible occurrences, impacts, and behaviors of cyberattacks (*e.g.*, DDoS, malware families, or a simple port scan). For that, *SecGrid* implements a set of components to process information from log files (*e.g.*, PCAP files) and present such information in a structured, interactive, and user-friendly way. Although *SecGrid* was designed for postmortem analysis of cyberattacks, real-time analysis is also possible by integrating real-time monitors with *SecGrid*. This depends especially on the complexity of the traffic (*e.g.*, entropy) and the time to process the information being provided, which can add a considerable overhead for scenarios that require real-time analysis in the current version of the platform.

In conclusion, *SecGrid* does provide the first integrated tool in support of a joint approach of processing, analyzing, and visualizing complex datasets of malicious traffic log information. The experiments conducted provided evidence of the benefits and scalability of the tool, such as very high usability to process and analyze the log files even to non-cyber security experts and efficiency to process a large amount of data. For the extraction and classification of cyberattacks, the *SecGrid* ML model obtained very high accuracy for the classification of DDoS attacks within the selected datasets, thus showing potential to apply the whole methodology (*e.g.*, development of miners, definition of the model, and implementation of the classification techniques) in scenarios with different cyberattacks.

Future work includes *(a)* the support of such real-time traffic by integrating *SecGrid* with different monitors and traffic files (*e.g.*, Netflow records and IoT sensors) and *(b)* the development of features that allow users to share their insights with others to support cybersecurity information sharing between interested stakeholders. Besides that, new miners and visualizations are planned to provide more accurate analysis for different scenarios and cyberattacks.

Acknowledgements

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] A. Kumar, T. Joon Lim, "Early Detection of Mirai-Like IoT Bots in Large-Scale Networks through Sub-sampled Packet Traffic Analysis," in *Advances in Information and Communication*, K. Arai and B. Rahul, Ed. Springer International Publishing, 2020, pp. 847–867.

[2] A. Parmisano, S. Garcia, M. J. Erquiaga, "Aposemat IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic," January 2020, https://www.stratosphereips.org/datasets-iot23, Last Visit May 2021.

[3] M. Bajer, "Building an IoT Data Hub with Elasticsearch, Logstash and Kibana," in *5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW 2017)*, Prague, Czech Republic, 2017, pp. 63–68.

[4] J. Brooke, "SUS: A Restropective," *Journal of Usability Studies (JUS)*, vol. 8, no. 2, pp. 29–40, 2013.

[5] Communication Systems Group CSG, "The SecGrid Project," 2021, http://www.csg.uzh.ch/csg/en/research/SecGrid, Last visit July 2021.

[6] R. J. Crouser, E. Fukuda, and S. Sridhar, "Retrospective on a Decade of Research in Visualization for Cybersecurity," in *IEEE International Symposium on Technologies for Homeland Security (HST)*, Waltham, USA, 2017, pp. 1–5.

[7] E. Kim , D. Gardner, S. Deshpande, R. Contu, D. Kish, C. Canales, "Forecast Analysis: Information Security, Worldwide, 2Q18 Update," September 2018, https://www.gartner.com/en/documents/3889055, Last Visit May 2021.

[8] M. Franco, J. V. der Assen, L. Boillat, C. Killer, B. Rodrigues, E. Scheid, L. Granville, and B. Stiller, "Poster: DDoSGrid: a Platform for the Postmortem Analysis and Visualization of DDoS Attacks," in *20th IFIP Networking (Networking 2021)*. Espoo, Finland: IFIP, June 2021, pp. 1–3.

[9] M. Franco, B. Rodrigues, and B. Stiller, "MENTOR: The Design and Evaluation of a Protection Services Recommender System," in *15th International Conference on Network and Service Management (CNSM 2019)*. Halifax, Canada: IEEE, October 2019, pp. 1–7.

[10] M. Franco, E. Sula, B. Rodrigues, E. Scheid, and B. Stiller, "ProtectDDoS: A Platform for Trustworthy Offering and Recommendation of Protections," in *Economics of Grids, Clouds, Systems, and Services*. Izola, Slovenia: Springer International, 2020.

[11] 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.

[12] K. A. Garcia, R. Monroy, L. A. Trejo, C. Mex-Perera, and E. Aguirre, "Analyzing Log Files for Postmortem Intrusion Detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1690–1704, 2012.

[13] H. Kang, D. Hyun Ahn, G. Min Lee, J. Do Yoo, K. Ho Park, H. Kang Kim, "IoT Network Intrusion Dataset," 2019, https://ieee-dataport.org/open-access/iot-network-intrusion-dataset, Last Visit May 2021.

[14] Z. He, T. Zhang, and R. Lee, "Machine Learning Based DDoS Attack Detection From Source Side in Cloud," in *IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud 2017)*, New York, USA, 2017, pp. 1–7.

[15] J. Hou, P. Fu, Z. Cao, and A. Xu, "Machine Learning based DDos Detection Through NetFlow Analysis," in *Military Communications Conference (MILCOM 2018)*, Los Angeles, CA United States, 2018, pp. 1–6.

[16] R. Poortinga, J. Ceron, J. Santanna, C. Hesselman, "European DDoS Clearing House Pilot," 2019, https://github.com/orgs/ddos-clearing-house, Last Visit May 2021.

[17] Raffael Marty, *Applied Security Visualization*. Addison-Wesley Professional, 2008.

[18] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A survey on machine learning techniques for cyber security in the last decade," *IEEE Access*, vol. 8, pp. 222 310–222 354, 2020.

[19] J. Suomalainen, A. Juhola, S. Shahabuddin, A. Mämmelä, and I. Ahmad, "Machine Learning Threatens 5G Security," *IEEE Access*, vol. 8, pp. 190 822–190 842, 2020.

[20] M. Suresh and R. Anitha, "Evaluating Machine Learning Algorithms for Detecting DDoS Attacks," *Communications in Computer and Information Science*, vol. 196, pp. 441–452, 2011.

[21] A. Thakkar and R. Lohiya, "Attack Classification using Feature Selection Techniques: a Comparative Study," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1249–1266, 2020.

[22] M. S. U. Banerjee, A. Vashishtha, "Evaluation of the Capabilities of Wireshark as a Tool for Intrusion Detection," *IEEE Transactions on Network and Service Management*, vol. 6, no. 7, pp. 1–5, 2010.

[23] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, "Scikit-Learn: Machine Learning Without Learning the Machinery," *GetMobile: Mobile Computing and Communications Review*, vol. 19, no. 1, p. 29–33, June 2015.

[24] Verizon, "Data Breach Investigations," March 2020, https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf, Last Visit May 2021.