Lessons Learned from Automated Sharing of Intrusion Detection Alerts: The Case of the SABU Platform

MARTIN HUSÁK, Masaryk University, Czech Republic
PAVOL SOKOL, Pavol Jozef Šafárik University, Slovakia
MARTIN ŽÁDNÍK and VÁCLAV BARTOŠ, CESNET, Czech Republic
MARTIN HORÁK, Masaryk University, Czech Republic

Sharing the alerts from intrusion detection systems among multiple computer networks and organizations allows for seeing the "big picture" of the network security situation and improves the capabilities of cyber incident response. However, such a task requires a number of technical and non-technical issues to be resolved, from data collection and distribution to proper categorization, data quality management, and issues of trust and privacy. In this field note, we illustrate the concepts and provide lessons learned on the example of SABU, an alert sharing and analysis platform used by academia and partner organizations in the Czech Republic. We discuss the initial willingness to share the data that was later weakened by the uncertainties around personal data protection, the issues of high volume and low quality of the data that prevented their straightforward use, and that the management of the community is a more severe issue than the technical implementation of alert sharing.

CCS Concepts: • **Security and privacy** → **Network security**; **Intrusion/anomaly detection and malware mitigation;**

Additional Key Words and Phrases: Cybersecurity, information sharing, intrusion detection, automation

ACM Reference format:
Martin Husák, Pavol Sokol, Martin Žádník, Václav Bartoš, and Martin Horák. 2023. Lessons Learned from Automated Sharing of Intrusion Detection Alerts: The Case of the SABU Platform. *Digit. Threat. Res. Pract.* 4, 4, Article 48 (October 2023), 11 pages.
https://doi.org/10.1145/3611391

1 INTRODUCTION

Collaboration and information exchange have been fundamental to cybersecurity since its foundations. The collaboration took various forms; warnings and announcements were distributed via mailing lists, public databases of vulnerabilities or cyber threat intelligence feeds emerged, and best practices in incident response were shared among the cybersecurity teams. A long-term trend is automating information exchange and structuring the information for their automated processing [9, 24, 27, 30]. The motivation for the automated information exchange was the perceived benefit of seeing the "big picture" of the cybersecurity situation and filling the blind spots in one organization with the data from the others in a timely manner. For example, the European Commission

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).
Authors' addresses: M. Husák and M. Horák, Institute of Computer Science, Masaryk University, Šumavská 416/15, 602 00 Brno, Czech Republic; emails: {husakm, horak}@ics.muni.cz; P. Sokol, Pavol Jozef Šafárik University in Košice, Jesenná 5, 040 01 Košice, Slovakia; email: pavol.sokol@upjs.sk; M. Žádník and V. Bartoš, CESNET Association of Legal Entities, Generála Píky 430/26, 160 00 Prague 6, Czech Republic; emails: {zadnik, bartos}@cesnet.cz.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2023 Copyright held by the owner/author(s).
2576-5337/2023/10-ART48
https://doi.org/10.1145/3611391

states in a working document [5] that insufficient information sharing on threats, risks, and incidents results in sub-optimal preparedness or response. The lack of data and information on computer systems and networks does not allow for conducting appropriate analysis and compiling statistics that could be used to raise awareness of the rising threats and to plan appropriate measures to tackle them. In the EU, the initiatives to share cybersecurity data are backed by the NIS Directive (EU 2016/1148). Similar initiatives can be found around the globe in national strategies and cybersecurity communities.

In this field note, we present the case study of the SABU[1] alert sharing platform that is operated in the academic computer network in the Czech Republic and is also used by peers from governments and industry. We first briefly introduce the topic of information sharing in the following subsection. Subsequently, we provide the description of the SABU platform and reasoning for the design choices in Section 2. In Section 3, we present the lessons learned from operating the platform and managing the community. Section 4 concludes the article.

1.1 Automated Information Exchange in Cybersecurity

Collaboration and information exchange in cybersecurity is as old as cybersecurity itself; the first mentions date back to the foundation of the notorious Morris worm and the foundation of CERT/CC in 1988. However, the first attempts at automating information exchange started much later with the development of the first **Collaborative Intrusion Detection Systems** (**CIDS**) [9]. Many technical issues and research challenges have emerged since then. For example, a need for a standardized format of shared information led to the development of IDMEF, IODEF, and many other formats [8]. Nevertheless, unstructured data-sharing platforms are still in use; an example is hpfeeds.[2]Simultaneously, the work on automated processing and analysis of shared security information started, and the process and fundamental algorithms were summarized by Valeur et al. [26]. Readers interested in a thorough review of collaborative intrusion detection are kindly referred to the book by Fung and Boutaba [9] or a survey by Vasilomanolakis et al. [27].

In the past decade, the focus of researchers and practitioners switched from CIDS to sharing platforms, such as open-source MISP,[3] which again brings novel research challenges and is a subject of ongoing development. MISP is a threat intelligence platform that originally responded mainly to the needs of cybersecurity forensics [29]. Its core functionality revolved around sharing **Indicators of Compromise** (**IoC**) and outputs of malware analysis, which is a novel use case in collaboration, substantially different from sharing timely security alerts. Over time, MISP evolved into a platform allowing for sharing, storing, correlating, and visualizing of structured IoCs related not only to forensics and threat intelligence but also targeted attacks, financial fraud information, vulnerability information, and even counter-terrorism information. MISP pushes the automation of data processing and offers metadata tagging and feeds for advanced threat intelligence in various open protocols and formats. Simultaneously, there are many emerging technologies based on STIX/TAXII standards[4] that cover sharing of both IoCs and alerts. In short, STIX is a description language, and TAXII is a transport protocol. However, particular sharing platforms and data feeds based on STIX/TAXII are operated by various entities, both open source and commercial, which may or may not comply with the needs of their users.

A complete list of alert sharing standards, tools, and platforms would be exhaustive, and thus, we only refer to selected works. Readers interested in detailed descriptions and comparisons of existing implementations are kindly referred to an older yet exhaustive report by ENISA [8], a research article by Steinberger et al. [24], a design consideration by Serrano et al. [22], or the recent work of Ramsdale et al. [21]. We also recommend the recent survey of cyber threat intelligence sharing by Wagner et al. [30].

[1]From the Czech title *Sdílení a Analýza Bezpečnostních Událostí*, translated as *Sharing and analysis of security alerts.*
[2]https://hpfeeds.org/
[3]https://www.misp-project.org/
[4]https://oasis-open.github.io/cti-documentation/

The cybersecurity community intuitively understands the importance of information sharing. Given the extensive number of platforms, standards, and tools [24, 27], it may seem that alert sharing is only a technical problem. In practice, however, many non-technical issues need to be addressed to make information sharing actually beneficial, ranging from motivation to share and building the community to the issues of trust and privacy and even legal issues. We will comment on the community issues in the case study. Trust issues are subject to ongoing research [22]. The legal issues come in the form of restriction and ensuring privacy when sharing potentially sensitive data [10, 25], but also in the form of stimulating and even enforcing data sharing as in the NIS directive. For example, several legal acts of the EU support the use of sharing platforms as forms of cooperation between the member countries, other countries, and organizations (e.g., FIRST[5]), and CSIRT/CERT teams.[6] According to the NIS2 directive proposal (Art. 26 (1) of the Proposal for a Directive of the EP and the Council on measures for a high common level of cybersecurity across the Union, repealing Directive (EU) 2016/1148), the EU member states are obliged to ensure that essential and vital entities may exchange relevant cybersecurity information, including information relating to threats, vulnerabilities, IoC, tactics, techniques and procedures, alerts and configuration tools.

2 CASE STUDY: SABU ALERT SHARING PLATFORM

To illustrate the concepts of security alert sharing in practice, we present an alert sharing platform called SABU.[7] The platform is based around a central hub and uses data enrichment, analysis, and reporting facilities. Using the taxonomy of CIDS [9] that is also applicable in this case, SABU is centralized, global, and information-based. Herein, we briefly introduce the history of SABU, present the technical background, and take a closer look at community policies, data analysis, and usage of the data.

The main difference between SABU and other platforms is that SABU follows the Unix principle of doing thing and doing it well. The components are mostly simple and focused on one activity (e.g., sending or receiving the data, performing one analytical task), which makes the whole platform modular and easy to deploy and maintain by both platform administrators and end users. However, the technical ease of use comes at the cost of the required effort in managing the user community. When comparing SABU to MISP, we may notice that data in SABU are exchanged promptly and not kept in history, which follows the primary motivation of both tools—SABU aims at prompt incident response, while MISP aims at threat intelligence and incident investigation.

2.1 Brief History of SABU and Warden

The foundation of the sharing platform can be traced back to late 2011 when CESNET and Masaryk University started working on Warden,[8] a tool for the collection and distribution of alerts that later became a core of the sharing platform. The first version of Warden was released in 2012; the current version has been stable since 2015. Simultaneously, the **Intrusion Detection Extensible Alert** (**IDEA**) format[9] was designed. In the early years, Warden was mostly an internal engineering project that was slowly developed and gained recognition. Several research works were published but have not received that much attention in the academic community. Nevertheless, Warden was mentioned in the ENISA documents [8] and was embraced by practitioners. As of 2022, the platform is still operated, and the data are shared within the community.

A major impulse was the funding of the SABU project awarded to CESNET and Masaryk University from 2016 to 2019. A large team started working intensively on a project of applied research funded by the Czech Republic. The team developed, released, and deployed a set of tools centered around Warden and built a dedicated

[5]Forum of Incident Response and Security Teams, https://www.first.org/
[6]Computer Security Incident Response Team, Computer Emergency Response Team.
[7]https://sabu.cesnet.cz/en/start
[8]https://warden.cesnet.cz/en/index
[9]https://idea.cesnet.cz/en/index

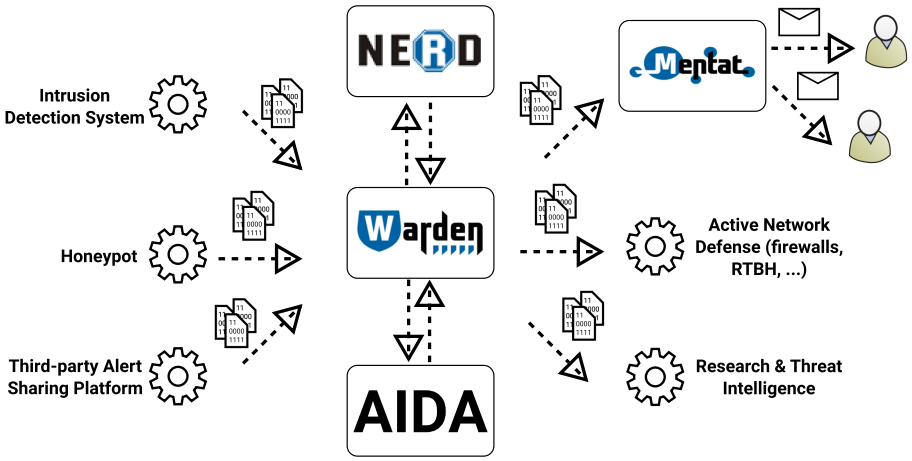

Fig. 1. Schema of SABU alert sharing platform.

community. The tools and approaches were also used and enhanced in the PROTECTIVE project funded by the EU.[10] The team members later published more research works [10, 13, 15–17, 25] and continued developing the tools that remained deployed and maintained by CESNET for use by the community. No more funding for research and development was asked for due to the achieved technological maturity and the lack of prospective research directions. Nevertheless, the former project investigators followed up their work in consecutive years and published a plethora of additional research articles [1, 3, 10–12, 19, 20, 23, 28, 32], datasets [18, 31], and tools [2, 14].

2.2 Technical Background

The scheme of the SABU alert sharing platform can be seen in Figure 1. SABU follows the Unix principle—each component does one thing and does it well. The central component is Warden, a hub that receives the alerts from sending connectors and distributes them to analytical and receiving connectors. Sending connectors provide alerts from data producers and other alert-sharing platforms. NERD[11] [2] is a reputation database that creates a profile of each reported IP address, enriches it with information from various external sources, and computes the address' overall reputation score. AIDA performs predictive data analysis [14]; it looks up frequent attack patterns, uses them to predict the next steps of the ongoing attacks, and sends the alerts of predictions back to Warden. The receiving connectors provide the data to consumers, e.g., active network defense appliances and data analysts. A special receiving connector is Mentat, a reporting tool that stores all the alerts, allows to search them, and, most importantly, notifies registered users by e-mail when an IP address from their constituency appears in the alerts. It can also create daily or weekly reports.

SABU uses the IDEA format to format the alerts. IDEA is inspired by IDMEF [7], but is customized to reflect operational needs, includes a taxonomy of security alerts, and prefers data serialization in JSON. Each message in IDEA format needs to include the IDEA version, a unique ID of a message in the platform, the time when the alert was raised, and an array of tags describing the type (category) of the alert. The most interesting pieces of information are to be found in *Source* and *Target* fields, where identifiers and types of attackers (*Source*) and victims (*Target*) are listed. The content and number of these entries are variable so that the message can indicate a range of events. Typically, *Source* and *Target* contain an IP address, hostname, URL, or e-mail address. *Node*

[10] https://cordis.europa.eu/project/id/700071
[11] https://nerd.cesnet.cz/

contains the identifier and description of the sensors, e.g., an intrusion detection system that sent the alert. The naming conventions in SABU allow for distinguishing sensors from different computer networks easily. There are also many other data fields for further details. Namely, the timing information is worth mentioning. While IDMEF requires a timestamp of the start of an event, IDEA requires a timestamp of its detection due to practical reasons. It is often difficult to determine or estimate the start of an event, especially when detecting network traffic anomalies or large-scale events. The timestamp of detection, in comparison, is always known.

The alerts are collected by peers, who use a plethora of heterogeneous sensors, such as intrusion detection systems and honeypots. SABU also receives security alerts from other sharing platforms and reporting tools, although this brings additional limitations, such as the incompleteness of the data, incompatible categorizations, and delays in delivery time. Due to the research interests of the founding members of the SABU community and other factors, the majority of data sources are flow-based network intrusion detection systems [4] that can be deployed in campus networks as well as on the backbone. Thus, there is a strong bias towards network security events in SABU, which has direct consequences on how the data can be used. Approximately 1.5 million alerts are generated and shared in SABU per day [18, 31].

2.3 Community and Policies

A cornerstone of an alert sharing platform is its community, which, in the case of SABU, is centered around the academic computer network in the Czech Republic. The intended scope of its usage is the computer network of CESNET, the Czech **national research and education network** (**NREN**), and its partners. Currently, CESNET operates a backbone network and connects 27 campus networks of universities and other research-related institutions such as hospitals, libraries, and state agencies. Over the years, non-academic partners joined the SABU community as well, including commercial Internet service providers and hosting services. It is worth mentioning that the Czech Republic is a relatively small country, so the cybersecurity community is not large; the people generally know each other and frequently meet in person, which facilitates collaboration. New peers are typically recruited via personal contact, and thus, trust among peers is implicit.

Joining the sharing platform with a new data source is accompanied by a quarantine policy. Before a new source of data is connected to SABU, it is routed to quarantine, a replica of Warden for testing purposes. A new node sends the data to the quarantine for at least two weeks while being manually inspected by platform operators, who may ask node administrators to change the reported alerts to comply with platform standards. Typical problems include a vague event description, misuse of tags, and inadequately brief or large messages. For example, a low-interaction honeypot LaBrea was deployed in over hundreds of IP addresses and was reporting every observed network connection to SABU as a network scanning attempt, which caused a flood of hundreds of thousands of alerts in the platform with close to no value. The honeypot was not allowed to send alerts to SABU until it was reconfigured to send only alerts of horizontal and vertical network scans of at least several network connections, which reduced the number of alerts to thousands per day and improved the readability of the alerts. Appropriate thresholds are another example of internal rules of the community, especially in the case of network scans. While some node operators set thresholds of connections for network scanning detection at their IDS to such low values as 3–10, IDS at backbone networks may have thresholds of thousands of connections to report network scanning events. Any analysis of shared alerts is complicated by a different understanding of the scale of particular events, which brings a need for establishing internal rules for alert sharing.

To further improve the quality of the data, two activities were further executed irregularly, feedback collection and investigation of false positives. Feedback collection aims at operators of receiving nodes. They were asked to fill in a form where they describe which shared data they are receiving, how they are using them, and how they are satisfied with their quality. The investigation into false positive alerts was an irregular activity triggered by either an unexpected, isolated incident or a long-term low satisfaction with the alerts by a particular sending node. Operators of the sending nodes are often unaware of false positives that are, in most cases, reported by receivers of the data, and cooperation is needed. For example, a university was informed in a shared alert of an

IP address from its IP address range that was attempting to connect to a honeypot of another peer. However, no traces of malicious traffic were found with regard to that IP address, and the attempts to connect to the honeypot seemed very unusual. Further inspection found that the honeypot IP address, for an unknown reason, probably a typo, appeared on a list of BitTorrent trackers, which caused connection attempts from legitimate hosts to the honeypot. These issues could have been investigated due to the feedback from a partner in a sharing platform.

2.4 Data Analysis

Analysis of shared alerts was performed in two directions, using either descriptive statistics or alert correlation. In the first case, we were most interested in the composition of the data, e.g., the number of alerts per type and top attacking IP addresses. Such analyses may be easily performed and visualized and thus are often displayed and published. Alert correlation, on the contrary, allows a deeper understanding of the shared data and more potential for practical use, although with increased complexity.

Descriptive statistics could be performed easily. Calculating the ratios and Top Ns of alert types, sources, attackers, victims, and other data elements, was beneficial for the administrators of the sharing platform as it told a lot about the platform. Continuous descriptive statistics may further detect changes and anomalies in total numbers and shares of alerts per type or sensor. In practice, this was used to detect a sensor failure, a sudden increase in a particular type of attack, and similar events. Nevertheless, descriptive statistics were of limited use for end users of sharing platforms as their outputs often could not be used to detect or mitigate a particular security event. An exception might be Top N lists, e.g., most frequent attacking IP addresses, from which we may derive a blocklist [3]. However, the frequent occurrence of a certain item, such as an IP address, in global scope does not necessarily mean that such an item can be observed in the computer networks of all participating parties.

Alert correlation aims at finding relations, patterns, and dependencies within the data, even the ones that are not clear at first sight. The procedures of alert correlation in SABU loosely followed the work of Valeur et al. [26] from 2004. Although the work is outdated in technical details, it provides an excellent high-level perspective on the process. The alerts first need to be preprocessed, i.e., converted to a unified format and syntactically filtered. Then, the alert fusion (or aggregation, deduplication) should be performed to remove duplicate entries and merge the alerts describing the same event. Such alerts may significantly skew any data analysis. This step may be surprising in terms of data volume reduction. For example, in the SABU platform, we described four simple scenarios of how duplicated alerts may appear in the platform, and an experiment showed that the aggregation (or deduplication) reduced the volume of the dataset by 85 % [17]. Correlating the alerts with the same source IP address over a period of time is referred to as thread reconstruction or attack session reconstruction and provides the sequence of actions the attacker has taken to perform an attack. The methods of sequential pattern mining proved beneficial in the case of SABU [12]. The other tasks described by Valeur et al., such as focus recognition and impact analysis, were not explored since they require knowledge of the target environments.

2.5 Distribution and Usage of the Data

Any peer may receive all the raw data shared within SABU. Alternatively, users may create a profile in the reporter (Mentat) and receive e-mail reports of the activities related to their computer networks immediately, as well as aggregated per a configurable time interval (e.g., daily). Such reports are informative and well-perceived in the community as they often describe events associated with a system in one organization but detected only by another organization. Our experience shows that the organization profile should include an alert prioritization schema that can decide if the alert should be reported immediately or reported later in the aggregate. In the case of an immediate report, it is also important to implement two thresholds—back-off and relapse. The back-off interval prevents reporting the same issue again (often, multiple similar alerts are shared due to the long-lasting nature of the issue), giving the administrator of the organization time to resolve the incident, while the relapse interval defines when the alerts should be forwarded again if the incident is not dealt with in due time.

Another use of shared raw data is creating a blocklist. The alerts contain identifiers, such as IP addresses and URLs, of malicious entities that can be denied access to the computer network. There are two issues with this use case. First, the alerts and their providers should be trustworthy. However, trust is highly subjective. While there are peers who use the data in this manner without any concerns, there are also peers who are extremely cautious and not willing to use them regardless of trust assurance. The second problem is the volume of the data. For example, 1.5 million alerts exchanged per day contain around 1 million unique IP addresses. Blocklisting all of them would quickly lead to the depletion of the firewall capacity or other means of traffic filtering. Moreover, typically only a fraction of the identifiers are of any use for a particular computer network, which makes the application of most of the rules futile. Therefore, the blocklists generated by analytical tools gained more popularity.

The research on the SABU project focused on the effective and timely use of the data. The most important decision was the move from a reactive to a proactive paradigm—instead of reacting to what the peers have observed, the goal was to estimate what could happen next and try to prevent it. Three approaches were designed to predict the attacks and forecast the overall security situation [12]. First, the frequent attack sequences were extracted from the data. If an attacker was observed to follow any known attack sequence, it was predicted that the remaining part of the sequence is likely to be taken by the attacker. Once a prediction is made, preventive measures can be applied to disrupt the ongoing attack [12, 14]. Second, a reputation database was built to keep the reputation scores of all the network entities (e.g., IP addresses) that appeared in the data shared in SABU. The reputation score is calculated from the frequency of appearance of an entity and the severity of incidents in which it participates. It is based on a machine-learning-based prediction of future attacks from the entity, so it can be used to generate an efficient predictive blocklist, listing only those entities that are likely to attack in the immediate future [1, 3]. Finally, time series analysis was used to forecast the overall situation, e.g., to predict the number of alerts, including their type and location, that will appear in the near future [20, 23].

3 LESSONS LEARNED

In this section, we present the lessons learned from the development and operation of the SABU platform. Several design considerations were already presented in the previous section. Herein, we confront our expectations with reality and highlight several key aspects that influenced the usability of SABU and the data sharing in it.

3.1 Community Management, Willingness to Share, and Legal Issues

Cybersecurity data are often perceived as private, and the general impression is that they are wanted by many but shared by a few. However, our experience was significantly different. Especially in academia, but also in other sectors, many peers offered their data out of enthusiasm, with only limited reciprocation in receiving the data. Most of the community members shared the data from their sensors as they intuitively understood the value of the shared data. At the same time, the practitioners did not attempt to receive, process, and use the shared data on their own but let the researchers explore it first. We assume this was caused by the lack of capacity to do so and the large volume of the data. Thus, the community involved more producers than consumers of the data. While in older works, the so-called free-riders, i.e., pure consumers, were perceived negatively [9], we actively sought out pure consumers as they were more likely to provide feedback on data quality and use cases of alert sharing.

The willingness to share changed dramatically in 2018 due to the implementation of the **General Data Protection Regulation** (**GDPR**) in the EU. The community became cautious of privacy and legal obligations. Rising fear, uncertainty, and doubt in the community were the reasons to perform a thorough legal analysis of privacy issues in alert sharing platforms. The analysis was performed to eliminate any doubts in the community, and it was found that sharing the security alerts outweighs the potential risks of disclosure of private information [10]. In retrospect, a thorough legal analysis was not needed, but it helped clarify the legal status of SABU. According to GDPR, it is necessary to consider the scope of data that can be considered personal data, the legal ground for

processing personal data for each activity, and the adoption of adequate security measures. The content of the shared data may vary from IoC and traffic captures to disk images of compromised systems. The IP addresses are vital for SABU. According to the Justice of the European Union cases, an IP address is personal data if the entity that processes it can attach additional data to it and thus clearly identify a natural person, which can be the case with managed IP addresses of SABU community members. The second aspect to consider is the legal ground for processing personal data. There are several cases in which processing shall be lawful (Art. 6 GDPR). One case represents legitimate interests pursued by the controller or a third party (Art. 6 (1c) GDPR). Recital 49 GDPR and CJEU case Breyer places activities related to protecting security within the framework of legitimate interests. Recital 69 of the NIS2 Directive proposal adds information sharing to the same framework [6]. One of the principles on which the protection of personal data in the GDPR is based is the protection by design and default (Art. 25 of the GDPR). According to this principle, the controller shall implement appropriate technical and organizational measures. The **Traffic Light Protocol** (**TLP**) is a recommended security measure (Rec. 6 of the NIS2 directive proposal) that can be applied on the level of the platform or at the level of individual nodes [10, 25]. The implementation of TLP was considered for SABU but not executed.

3.2 Data Composition and Quality

The repeated statistical analysis of the composition of the data provided important information on the quality of the data. For example, in the beginning, we found that the vast majority of alerts (over 95 %) were of type *Recon.Scanning*, which is used for labeling network scanning events. Furthermore, about half of such alerts were raised by a single sensor. Such a bias towards a certain type of security event was not desired, so the community focused on delivering more diverse alert types; namely, the detection of brute-force password attacks gained popularity. The most active sensor was found to be reporting events of very fine granularity in comparison to similar sensors, so it was refactored accordingly. The overall quality of the data in the sharing platform was raised continuously, and in the end, the ratio of reported network scans was around 80 %, while the ratio of reported brute-force password attacks increased to about 15 %.

The composition of the data and the willingness to share them is worth mentioning. Network scanning is indeed prevalent in what we can observe in the network traffic. It is fairly easy to detect, there is a low risk of false positive detection, the severity of such an action is low, and it does not disclose anything about the target computer network. There are no issues for the contributors in sharing such alerts. A similar situation is with brute-force password attacks that can also be detected with high confidence and using a minimal amount of information about the target. Therefore, the contributors typically shared all the alerts of these two types. On the contrary, the contributors were not willing to share the alerts of other types of attacks because they were not confident in their sensors; they were afraid of sharing a false positive or disclosing too much about their computer network. Thus, the original bias toward network-based detection was further amplified towards only a few alert types.

Moreover, community management efforts toward increasing data quality are a double-edged sword. Since many peers shared their data out of enthusiasm and perceived benefits and not because of obligation, they may stop at any time. Any demand on reconfiguring or tuning their sensors towards higher data quality may be dismissed, and the peers may decide to stop sharing instead of obeying the regulations and recommendations. This is especially applicable to smaller or understaffed organizations in which the cybersecurity personnel does not have the free capacity or required knowledge.

3.3 Data Analysis and Usage

The prediction of the AIDA framework based on sequential rule mining and attack projection showed interesting results with a high prediction accuracy of 60–90 % [12]. Nevertheless, two issues became apparent. First, many frequent sequences contained only network scanning activities due to the lack of other alert types. It was still

interesting to see patterns such as "the attacker first scans computer network X, then computer network Y, and finally computer network Z" and use the observations in computer networks X and Y to prevent scanning computer network Z. More heterogeneous attack sequences were found with the rising numbers of brute-forcing alerts, but the vast numbers of network scanning alerts still caused the top frequent sequences to consist mostly of scanning activities. Second, even though the peers could easily receive the predictions from SABU, they were initially reluctant to use them. The predictions were labeled as experimental so that the users could filter them out as they did not consider them trustworthy. They had to be persuaded that the predictions are based on recent observations and that no predictions could be made about network entities (e.g., IP addresses) that did not behave maliciously in the recent past. From a technical perspective, it was the right choice to use modern approaches to streaming data analysis (AIDA is based on Apache Spark, Apache Kafka, and Esper tools [14]); it fastened the developments and enabled real-time processing of big data that the shared alerts are. The drawback was a complicated local deployment for testing and research purposes.

The reputation database NERD became successful on its own as the developers included more data sources (e.g., public blocklists and other sharing systems) and provided public access to the data about the history of an entity, both via a web interface for human users and an API for automated services [2]. The reputation scores and contextual data are used by practitioners to promptly assess the risks associated with a network entity and facilitate incident triage. Some users also check NERD periodically to see if there are any IP addresses from their own network—an indication they might be compromised. The database can also be used to generate blocklists of the worst offenders [3]. Such blocklists achieve high accuracy because the worst offenders typically act maliciously on a daily basis for several weeks. An important practical aspect is that the blocklist can be of an arbitrary length so that the user may use a blocklist that fits the capacity of their firewalls and other devices.

The time series forecasting provided a more high-level, strategic perspective on the network security situation. The partial time series consisting of qualitative components (alert category, network protocol, and port number) were found to be more useful than the time series of the total number of alerts as they are capable of predicting the number of alerts or vulnerabilities the security team will have to process or sudden increase of malicious activity related to certain port or service [12, 20]. Various analyses and experiments [12, 23] showed comparable results for different methods. It is also advisable to consider the combination of several methods and the use of neural networks. The rolling window approach is advisable due to the large volume of the data. This approach means that each round of the evaluation model removes the oldest observation from the training set and, at the same time, adds one new observation from the test set to the training set. The methods that used the rolling window have comparable results to those without this approach [20]. This result is essential for forecasting based on time series without the need for increasing memory requirements [12]. Another aspect of the time series analysis is the parameters' setting. A combination of 30 minutes, seven days, and one month of data was shown to be most suitable for the outlined use cases [20]. The 30-minute time units proved suitable for predicting alert rates, while the 3-day time units excelled in predicting vulnerabilities. Nevertheless, outputs of such high-level analyses are yet to be adopted by practitioners, who are often capable of processing only low-level actionable pieces of information. They may also struggle with the proper setting of method parameters, so such analyses are advised to be conducted centrally or as a service.

4 CONCLUSION

In this article, we recapitulated the history of the SABU alert sharing platform and presented the lessons learned from its development and operation. The sharing platform is run and used by the community of Czech academic institutions and partners from the government, industry, and abroad. However, despite the generally perceived benefits of information sharing and the research and development efforts, there are many limitations to the effective use of the platform. We do not perceive any major benefits of SABU compared to other platforms in terms of technology, but we learned valuable lessons about using the data and managing the user community. The most important takeaways are as follows:

— Contrary to the expectations and experience in the cybersecurity community, we did not face unwillingness to share the data (except for a period of uncertainty related to personal data protection, including the introduction of the GDPR [10, 25]); we instead saw volunteers contributing with data and even tools to share them more effectively.
— The number of enthusiastic contributors generated vast amounts of low-quality data heavily biased towards a small number of alert types [18, 31], namely, those that can be easily detected with a low false positive rate.
— Improving the quality of the data was a long and difficult task with lesser impact than expected. Contributors may also stop sharing instead of conforming to additional rules because of the lack of capacity.
— The data analysis was prone to the garbage-in garbage-out principle, and the utilization of its results remained relatively low. The researchers had to shift from a reactive to a proactive paradigm to come up with ways to use the data effectively and in a timely manner [3, 12, 14].
— Although a lot of effort was put into providing means to use the shared data to protect computer networks from outer threats, probably the most used and appreciated use case is the one in which a misbehaving device is found in a computer network thanks to a sensor in another computer network, which is automatically reported to the administrator of the former one.

Nevertheless, we believe all the efforts generated valuable knowledge and experience that qualitatively enhance current research and development in cybersecurity. Namely, we learned there are large differences between organizations, their computer networks, sensors, and the data they produce. Such knowledge is not only useful for future data-sharing attempts but also in alert correlation, analysis of advanced attacks, and network forensics. Moreover, we approached the topics of data quality or willingness to share the data, which were underrepresented in previous works on the topic [9, 24, 27, 30].

We plan to further pursue the goals of SABU, analyzing the data and maintaining the platform. Although the tools used to support SABU are publicly available and free to use, we do not intend to grow the user base to a larger extent to preserve the implicit trust within the community. Nevertheless, a larger-scale deployment with no implicit trust between the peers, which would call for revisiting the guidelines and researching the topic of trust, would be an interesting direction for future work.

REFERENCES

[1] Mohammad Samar Ansari, Václav Bartoš, and Brian Lee. 2022. GRU-based deep learning approach for network intrusion alert prediction. *Future Generation Computer Systems* 128 (2022), 235–247.
[2] Václav Bartoš. 2019. NERD: Network entity reputation database. In *Proceedings of the 14th International Conference on Availability, Reliability, and Security (ARES'19)*. ACM, 7 pages.
[3] Václav Bartoš, Martin Žádník, Sheikh Mahbub Habib, and Emmanouil Vasilomanolakis. 2019. Network entity characterization and attack prediction. *Future Generation Computer Systems* 97 (2019), 674–686.
[4] Tomas Cejka, Vaclav Bartos, Marek Svepes, Zdenek Rosa, and Hana Kubatova. 2016. NEMEA: A framework for network traffic analysis. In *Proceedings of the 2016 12th International Conference on Network and Service Management (CNSM'16)*. IEEE, 195–201.
[5] European Commission. 2013. COMMISSION STAFF WORKING DOCUMENT IMPACT ASSESSMENT Accompanying the document Proposal for a Directive of the European Parliament and of the Council Concerning measures to ensure a high level of network and information security across the Union. (2013). Retrieved on April 11, 2023 from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52013SC0032
[6] Andrew Cormack. 2021. NISD2: A common framework for information sharing among network defenders. *SCRIPTed* 18, 1 (2021), 16 Pages.
[7] H. Debar, D. Curry, and B. Feinstein. 2007. The Intrusion Detection Message Exchange Format (IDMEF). RFC 4765 (Experimental). (2007). Retrieved April 11, 2023 from http://www.ietf.org/rfc/rfc4765.txt
[8] ENISA. 2014. Standards and tools for exchange and processing of actionable information. (2014). Retrieved April 11, 2023 from https://www.enisa.europa.eu/publications/standards-and-tools-for-exchange-and-processing-of-actionable-information/at_download/fullReport
[9] Carol Fung and Raouf Boutaba. 2013. *Intrusion Detection Networks: A Key to Collaborative Security*. CRC Press, Boca Raton, FL.

[10] Martin Horák, Václav Stupka, and Martin Husák. 2019. GDPR compliance in cybersecurity software: A case study of DPIA in information sharing platform. In *Proceedings of the 14th International Conference on Availability, Reliability, and Security (ARES'19)*. ACM, 8 pages.
[11] Martin Husák, Tomáš Bajtoš, Jaroslav Kašpar, Elias Bou-Harb, and Pavel Čeleda. 2020. Predictive cyber situational awareness and personalized blacklisting: A sequential rule mining approach. *ACM Transactions on Management Information Systems* 11, 4, (2020), 16 pages.
[12] Martin Husák, Václav Bartoš, Pavol Sokol, and Andrej Gajdoš. 2021. Predictive methods in cyber defense: Current experience and research challenges. *Future Generation Computer Systems* 115 (2021), 517–530.
[13] Martin Husák and Jaroslav Kašpar. 2018. Towards predicting cyber attacks using information exchange and data mining. In *Proceedings of the 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC'18)*. IEEE, 536–541.
[14] Martin Husák and Jaroslav Kašpar. 2019. AIDA framework: Real-time correlation and prediction of intrusion detection alerts. In *Proceedings of the 14th International Conference on Availability, Reliability, and Security (ARES'19)*. ACM, pages.
[15] Martin Husák, Jaroslav Kašpar, Elias Bou-Harb, and Pavel Čeleda. 2017. On the sequential pattern and rule mining in the analysis of cyber security alerts. In *Proceedings of the 12th International Conference on Availability, Reliability, and Security*. ACM, Reggio Calabria, 22:1–22:10.
[16] Martin Husák and Milan Čermák. 2017. A graph-based representation of relations in network security alert sharing platforms. In *Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM'17)*. IEEE, Lisbon, 891–892.
[17] Martin Husák, Milan Čermák, Martin Laštovička, and Jan Vykopal. 2017. Exchanging security events: Which and how many alerts can we aggregate?. In *Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM'17)*. IEEE, Lisbon, 604–607.
[18] Martin Husák, Martin Žádník, Václav Bartoš, and Pavol Sokol. 2019. Dataset of intrusion detection alerts from a sharing platform. (2019). Retrieved April 11, 2023 from https://data.mendeley.com/datasets/p6tym3fghz/1
[19] Arbnor Imeri and Ondrej Rysavy. 2023. Deep learning for predictive alerting and cyber-attack mitigation. In *Proceedings of the 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC'23)*. IEEE, 0476–0481.
[20] Patrik Pekarčík, Andrej Gajdoš, and Pavol Sokol. 2020. Forecasting security alerts based on time series. In *Proceedings of the Hybrid Artificial Intelligent Systems*. Springer International Publishing, Cham, 546–557.
[21] Andrew Ramsdale, Stavros Shiaeles, and Nicholas Kolokotronis. 2020. A comparative analysis of cyber-threat intelligence sources, formats, and languages. *Electronics* 9, 5 (2020), 22.
[22] Oscar Serrano, Luc Dandurand, and Sarah Brown. 2014. On the design of a cyber security data sharing system. In *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security (WISCS'14)*. ACM, 61–69.
[23] Pavol Sokol, Richard Staňa, Andrej Gajdoš, and Patrik Pekarčík. 2023. Network security situation awareness forecasting based on statistical approach and neural networks. *Logic Journal of the IGPL* 31, 2 (2023), 352–374.
[24] Jessica Steinberger, Anna Sperotto, Mario Golling, and Harald Baier. 2015. How to exchange security events? Overview and evaluation of formats and protocols. In *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 261–269.
[25] Václav Stupka, Martin Horák, and Martin Husák. 2017. Protection of personal data in security alert sharing platforms. In *Proceedings of the 12th International Conference on Availability, Reliability, and Security*. ACM, Reggio Calabria, 65:1–65:8.
[26] Fredrik Valeur, Giovanni Vigna, Christopher Kruegel, and Richard A. Kemmerer. 2004. Comprehensive approach to intrusion detection alert correlation. *IEEE Transactions on Dependable and Secure Computing* 1, 3 (2004), 146–169.
[27] Emmanouil Vasilomanolakis, Shankar Karuppayah, Max Mühlhäuser, and Mathias Fischer. 2015. Taxonomy and survey of collaborative intrusion detection. *ACM Computing Surveys* 47, 4 (2015), 33 pages.
[28] Samuel Šulan and Martin Husák. 2022. Limiting the size of a predictive blacklist while maintaining sufficient accuracy. In *Proceedings of the 17th International Conference on Availability, Reliability, and Security (ARES'22)*. ACM, 6 pages.
[29] Cynthia Wagner, Alexandre Dulaunoy, Gérard Wagener, and Andras Iklody. 2016. MISP: The design and implementation of a collaborative threat intelligence sharing platform. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security (WISCS'16)*. ACM, 49–56.
[30] Thomas D. Wagner, Khaled Mahbub, Esther Palomar, and Ali E. Abdallah. 2019. Cyber threat intelligence sharing: Survey and research directions. *Computers and Security* 87 (2019), 101589.
[31] Jan Wrona. 2021. A Week-Long Capture Of 8 Million Intrusion Detection Alerts Obtained Via an Alert Sharing Platform Warden. (2021). Retrieved April 11, 2023 from https://zenodo.org/record/4683701
[32] Martin Zadnik, Jan Wrona, Karel Hynek, Tomas Cejka, and Martin Husák. 2022. Discovering coordinated groups of IP addresses through temporal correlation of alerts. *IEEE Access* 10 (2022), 82799–82813.

Received 15 July 2022; revised 1 May 2023; accepted 24 July 2023