Towards Predicting Cyber Attacks Using Information Exchange and Data Mining

Martin Husák, Jaroslav Kašpar
Institute of Computer Science & Faculty of Informatics,
Masaryk University, Brno, Czech Republic
{husakm, kaspar}@ics.muni.cz

***Abstract*—In this paper, we present an empirical evaluation of an approach to predict attacker's activities based on information exchange and data mining. We gathered the cyber security alerts shared within the SABU platform, in which around 220,000 alerts from heterogeneous geographically distributed sensors (intrusion detection systems and honeypots) are shared every day. Subsequently, we used the methods of sequential rule mining to identify common attack patterns and to derive rules for predicting attacks. As we illustrate in this paper, a collaborative environment allows attack prediction in multiple dimensions. First, we can predict what will the attacker do next and when. Second, we can predict where will the attack hit, e.g., when an attacker is targeting several networks at once. In a week-long experiment, we processed in total over 1 million alerts, from which we mined predictive rules every day. Our findings show that most of the rules display stable values of support and confidence and, thus, can be used to predict cyber attacks in consecutive days after mining without a need to actualize the rules every day.**

***Keywords*—Attack prediction, Collaborative security, Information exchange, Data mining**

I. Introduction

Collaboration has become an important topic in cyber security. Sharing the knowledge, experience, and timely information on current threats and attacks is regarded as a crucial part of operations of cybersecurity teams. Sharing of timely information on current threats and attacks seems as a promising form of early warning, which can be used to take preemptive measures to defend a network. It is no surprise that both researchers [1–3] and practitioners [4,5] have taken the initiative to create methods and build tools that would allow automated security information exchange and use them to increase the protection of the networks.

One of the promising use cases of security information exchange is predicting cyber attacks [3]. In the past, we have seen many attempts to predict the upcoming attacks or attack steps [6]. However, the majority of such methods faced serious problems that caused these methods to never reach a deployment in production. For example, they required a library of attack descriptions or an observation point with very detailed visibility into the network traffic and highly precise intrusion detection. It was, and still is, highly demanding to have a single observation point in the network that could provide every piece of information needed to detect and successfully predict the next event in an ongoing attack. However, collaboration seems to make the problem potentially easier to approach. First, many attacks, such as malware infections, were observed to happen in the same fashion in close time and spanning multiple networks [1]. Thus, we can use observations from a neighbor network to predict upcoming events in our network, instead of relying solely on own intrusion detection systems. Second, heterogeneity of intrusion detection systems may provide a complex view on an attack even though the individual intrusion detection systems do not have complete detection capabilities and, thus, may complement each other [2].

To formalize the scope of our work, we state three research questions which we shall answer: i) How to learn about cyber attacks from shared alerts? ii) Which attacks or attack steps can be predicted with sufficient accuracy? iii) Are the attack patterns stable over time? First, we need to learn about the cyber attacks and common patterns in them, so that we may adequately represent them and set up rules for predicting the next move of an attacker. For this purpose, we briefly introduce the security alert sharing platform SABU [7], where partners from academia and industry may share alerts from their intrusion detection systems. We then apply data mining (specifically, sequential rule mining) methods on these data to obtain the attack patterns. The outputs, referred as sequential rules, can be immediately used for predicting future cyber attacks. Second, we discuss the accuracy of attack prediction that we can also infer from the sequential rules. Finally, we are interested whether the findings from a single data mining session may be applied in consecutive days.

The contribution of this paper is an introduction of cyber attack predictions into the domain of collaborative environment and illustration of unique traits that such environment offers for this task. Our experiment demonstrates how does the state-of-the-art data mining methods perform on real-world cybersecurity alerts regarding the soundness of results and outlines the foundations for future work and practical implementations.

This paper is organized into seven sections. After the introduction, we sum up the related work in Section II. The SABU alert sharing platform is described in Section III. Section IV discusses mining patterns in the attacks observed in the collaborative environment. The experiment setup is outlined Section V and the results are presented and discussed in Section VI. Finally, Section VII concludes the paper and outlines future work.

II. Related Work

Collaboration and information exchange has become a crucial part of cybersecurity practice. Research and development have focused on automating the process so that timely and important pieces of information would be exchanged in order to provide early warning, increase the precision of intrusion detection or simply to announce a threat. From a research perspective, a lot of attention was dedicated to collaborative intrusion detection, which is essentially a low-level information sharing between intrusion detection systems. The theoretical background to this topic was described in details by Fung and Boutaba [2], and surveyed by Zhou et al. [1] and Vasilomanolakis et al. [3], who also state collaborative attack prediction as an open problem. From the practitioner's perspective, many implementations of security information sharing platform exist as surveyed by ENISA [4,5]. Real-world information exchange platforms focus rather on high-level pieces of information such as security alerts, formal representations of security events reported by intrusion detection systems. An overview of formats and protocols for exchange of such information was presented by Steinberger et al. [8].

Formalization of security alerts allowed a whole new field of research to emerge, the security alert correlation [9]. The task of alert correlation is to put together corresponding alerts, find relations between alerts, and reconstruct the progress of an attack, but also to recognize the focus of an attacker and analyze the impact of potential security incidents. The detailed tasks and processes of security alert correlation were proposed by Valeur et al. [10], while Cuppens and Miege [11] discussed the problem from the perspective of collaborative intrusion detection. Briefly, the alerts from sensors need to normalized, fused (aggregated), and verified first. Subsequently, the attack session can be reconstructed for the purposes of attack focus recognition, multi-stage attack correlation, and impact analysis [10]. The three later stages of alert correlation overlap with the task of attack prediction.

When the security alerts are correlated, and certain relations between them are observed, we can use this knowledge to predict the behavior of future attackers and progress of future attacks. In the past, we have seen attempts to predict the attack progression or the desired goal of an adversary (this is often referenced in literature as *attack projection* and *focus recognition*). Early approaches used predefined models of attack scenarios. Such models could be attack graphs [12–14], Bayesian networks [6], or Markov models [15,16], to cite the most relevant contributions. If a series of detected events corresponds to a part of an attack scenario in the model, the remaining parts of the scenario can be predicted. However, due to high demands on creating such models manually and continuously changing threat landscape, researchers started using methods of data mining to (semi-)automatically create the patterns and models to match and project running attacks [13,14,16–18]. Farhadi et al. [16] proposed a real-time approach, Kim and Park [17] proposed using continuous data mining, and Jiang et al. [18] combined data mining with similarity search. Simultaneously, attack prediction methods based on machine learning were proposed [19,20].

Although the attempts to predict attacks originate in the early 2000s, they still mostly focus on predicting the events for a single observation point or a single network and were rarely combined with collaborative approaches to cyber security and intrusion detection. Only a few works were found to approach the problem, although with rather simple approaches. For example, collaborative predictive blacklisting has been a subject of research [21,22] with promising results against specific attacks. However, a generic approach similar to early attempts to attack prediction is still an open research problem [3]. Further, many of the proposed attack prediction methods have only been evaluated using the datasets, but we know little about running such methods on data from real networks [23]. In a related domain of threat intelligence, Fachkha et al. [24] used data mining on darknet data to predict cyber threats in the global scope.

III. SABU Alert Sharing Platform

To describe the requirements and operational environment of our approach to attack prediction, we briefly describe the alert sharing platform SABU [7], which we used as a framework for our experimental evaluation. SABU is a platform for sharing and analysis of security alerts developed by CESNET and Masaryk University. The intended scope of its usage is the network of CESNET, Czech national research and education network, and its partners. Currently, CESNET operates a backbone network and connects 27 campus networks. The backbone and campus networks are monitored by heterogeneous intrusion detection systems, which generate alerts shared in SABU. The majority of detectors use flow-based network monitoring [25] and flow-based intrusion detection [26], which can be deployed in campus networks as well as on the backbone. SABU also receives many alerts from honeypots deployed in the participating networks. Warden [27], a hub for alert sharing and a key component of SABU platform, is published as open-source, along with SABU connectors for popular honeypots, such as *Dionaea*, *Kippo*, and *Cowrie*. The analytical components, however, are yet to be published.

A typical life-cycle of a shared security event looks as follows. A sensor, i.e., any intrusion detection system or a honeypot, in a participating network detects a security event and raises an alert. Since the sensor is connected to SABU platform, the alert is sent to the central hub, Warden [27], which allows sharing the alert with other peers and also passes the alert to the centralized alert processing facilities, where the alerts are processed. The processing consists of syntactic and semantic checks, alert aggregation and correlation [28], visualization [29], and other analytics, such as attack prediction, which is proposed in this paper.

The SABU platform uses Intrusion Detection Extensible Alert (IDEA) format for exchanging data [30]. IDEA is inspired by IDMEF [31], but is customized to reflect operation needs, includes a taxonomy of security events, and prefers data serialization in JSON. The most important items in an IDEA record, for the purpose of this paper, are *Category*, *Source.IP*, *Node.Name*, and *DetectTime*. *Category* lists the type of an event, e.g., network scanning or exploitation attempt. The full list of data entries and event categories is available at the IDEA website [30]. *Source.IP* is an IP address of an event originator, typically the attacker. In the field of network security, we mostly work with IP addresses only, so we did not consider other *Source* fields to be applicable for this paper. *Node.Name* tells us the name of a sensor, e.g., an intrusion detection system or a honeypot that observed and shared the event described in the alert. The naming conventions in SABU platform allows us to easily distinguish sensors from different networks. Finally, *DetectTime* is a mandatory field that describes the timestamp of the event's detection (contrary to event's starting time, which is not always discoverable).

IV. Mining the attack prediction rules

To successfully predict the next step of an attack, we first need to know which attacks are there in the network that can be observed, formalized, matched, and projected. The main disadvantage of previous attempts to attack prediction discussed in related work (e.g., [6,12,15]) was their reliance on an attack library, i.e., a database of possible attacks described in details in machine-readable format. Such library is very hard to create and maintain manually as the threat landscape is continuously evolving in a rapid pace, the detection capabilities in the network might be insufficient to capture all the security events, and even the set of expected attacks may not overlap with the set of actual ongoing attacks. However, data mining has been shown to be a suitable method of getting an insight into what is actually happening in the network and to (semi)automatically build the prediction models [13–17].

In our previous work [23], we have discussed sequential pattern and rule mining methods applied to the analysis of cyber security alerts. The literature search and experimental evaluation led us to the conclusion that sequential pattern mining is applicable in the correlation of security alerts, while sequential rule mining can also be applied in attack prediction. Herein, we extend our previous findings considering the attack prediction use case.

To run sequential rule mining, we first need to create a sequence database, that stores all the sequences found in the input data [32]. Throughout this papers, the database entries are inferred from IDEA records shared within SABU using *Source.IP* as a key. Items in a sequence consist of a structure containing *Node.Name*, *Category*, and *Target.Port*. *DetectTime* is used for sorting the items in a sequence. As we can see, the sequences describe actions of an attacker behind a single IP address. Other combinations of keys and items were discussed in the previous work [23]. It is worth noting that sequential rule mining does not, by default, work with time differences between items.

By running sequential rule mining over security alerts obtained from SABU, we obtained numerous sequential rules that can be used for attack prediction. Herein, we show two examples of mined rules to illustrate commonly found rules and their important parameters, support and confidence. Briefly, support (herein referenced as #SUPP) indicates how many times the sequence has been observed in the dataset, divided by the total number of sequences. Confidence (referenced as #CONF in the examples) indicates how often the rule has been found to be true, which can be directly used as a probability value for future predictions. The items in the sequences consist of three parts, identifier of the intrusion detection system, type of an event (according to IDEA taxonomy [30]), and TCP port number, if available. The intrusion detection system identifier usually includes the name of the software and the name of the organization, where is it deployed. Due to privacy reasons, we anonymized the organization names throughout the paper.

The first example illustrates spread of an attack as captured by popular SSH honeypots Kippo and Cowrie on TCP port 22 in the networks of three distinct organizations:

```
Organization_A.kippo:Attempt.Login:22,
Organization_B.cowrie:Attempt.Login:22
=> Organization_C.kippo:Attempt.Login:22
#SUPP: 0.00367 #CONF: 0.54545
```

The mined sequential rule indicates that if an attacker attempted brute-force attack (*Attempt.Login*) on a honeypot in organizations A and B, the attacker will also brute-force a honeypot in organization C with approximately 54 % confidence (distilled from #CONF value). Such sequences hold for 0.37 % of all sequences in the dataset (distilled from #SUPP value). Similar rules were found for other combinations of honeypots from these organizations, although with lower support and confidence values. Thus, we can infer that the attackers typically follow an order of targets given by the rule in the example.

The second example illustrates the steps of an attack against a single target in two mined rules:

```
1) Organization_A.dionaea2:Recon.Scanning:139
   => Organization_A.dionaea2:Attempt.Exploit:445
   #SUPP: 0.00613 #CONF: 0.83333
2) Organization_A.dionaea1:Recon.Scanning:139
   => Organization_A.dionaea1:Attempt.Exploit:445
   #SUPP: 0.00551 #CONF: 0.9
```

Both rules describe a situation, where an attacker scans the TCP port 139 and subsequently attempts to exploit a network service on port 445. Both events are captured by a Dionaea honeypot. This is a common scenario of exploitation of SMB/NetBIOS network services. Very high confidence value suggests that the exploitation attempts are very likely once a network scanning is observed. Similar confidence values in rules mined from outputs of two distinct honeypots also suggest that the rule can be generalized for any similar target.

V. Experiment Setup

As we have seen in the examples in the previous section, sequential rule mining can discover various attack patterns. In practice, however, the results may be influenced by misleading inputs, such as duplicated entries and false positives [28]. Thus, we designed an experiment to empirically evaluate the use of sequential rule mining for attack prediction. The Top-K rules are the ones with confidence value higher than a threshold and with the highest support value. In the experiment, we

TABLE I
TOP-10 SEQUENTIAL RULES FROM THE FIRST DAY WITH SUPPORT, CONFIDENCE, AND MINIMAL AND AVERAGE TIME DIFFERENCES (IN SECONDS)

Rule	Input		Output	Support	Confidence	Min. Δt	Avg. Δt
1	Org_A.tarpit:Recon.Scanning:2323, Org_A.nemea.hoststats:Recon.Scanning::None	⇒	Org_A.tarpit:Recon.Scanning:23	0.00438	0.88386	12	1,530
2	Org_A.nemea.bruteforce:Attempt.Login:23	⇒	Org_A.tarpit:Recon.Scanning:23	0.00824	0.53465	121	7,539
3	Org_A.nemea.hoststats:Recon.Scanning:None	⇒	Org_A.hoststats:Recon.Scanning:None	0.01987	0.68214	1	401
4	Org_A.tarpit:Recon.Scanning:2323	⇒	Org_A.tarpit:Recon.Scanning:23	0.06655	0.70099	901	5,882
5	Org_A.tarpit:Recon.Scanning:2222	⇒	Org_A.tarpit:Recon.Scanning:22	0.00834	0.58155	914	7,041
6	Org_A.tarpit:Recon.Scanning:2323, Org_A.hoststats:Recon.Scanning:None	⇒	Org_A.tarpit:Recon.Scanning:23	0.00487	0.89071	21	2,019
7	Org_A.nemea.hoststats:Recon.Scanning:None, Org_B.nemea.hoststats:Recon.Scanning:None	⇒	Org_A.hoststats:Recon.Scanning:None	0.00544	0.80088	4	735
8	Org_A.hoststats:Recon.Scanning:None, Org_A.tarpit:Recon.Scanning:443	⇒	Org_A.tarpit:Recon.Scanning:80	0.00289	0.90000	35	22,754
9	Org_A.hoststats:Recon.Scanning:None, Org_B.nemea.hoststats:Recon.Scanning:None	⇒	Org_A.nemea.hoststats:Recon.Scanning: None	0.00411	0.60284	1	2,698
10	Org_A.tarpit:Recon.Scanning:2323, Org_A.hoststats:Recon.Scanning:None, Org_A.nemea.hoststats:Recon.Scanning:None	⇒	Org_A.tarpit:Recon.Scanning:23	0.00266	0.83962	12	1,528

used real data from the SABU alert sharing platform presented earlier in this paper.

Top-K sequential rule mining algorithm TopKRules [32], implemented in the SPMF library [33], was used to mine the rules. Two parameters were applied to filter the results. First, we set $K = 10$ to get the Top-10 rules with the highest support. Second, we set a threshold to the confidence value to 0.5. Otherwise, we followed the data processing procedure presented in Section IV.

The plan of the experiment goes as follow. First, we run sequential rule mining over the security alerts reported in one day and keep the Top-10 rules. Subsequently, we repeat Top-10 sequential rule mining in several consecutive days and compare the results with the findings from the first day. If the same rule was found in several consecutive days, we analyze its support and confidence values to see if they appear to be stable over time, or if there are anomalous rules appearing in only a few days. Further, we use the confidence values of the same rule from different days to estimate the success rate of attack predictions based on that rule.

VI. Results and Discussion

In this section, we present and discuss the results of the proposed experiment. The data retrieval lasted for five days. In total, we processed 1,108,204 security alerts, around 220,000 alerts per day. Each day, we retrieved around 130,000 sequences from the alerts reported that day.

Table I shows the Top 10 rules mined during the first day of the experiment. These rules are left deliberately unfiltered to illustrate the actual results of data mining over real-world security alerts, including any potential false positives and distortions. It is apparent that most of the rules contain alerts of network scanning generated by a small number of intrusion detection systems from only two organizations. The explanation behind this fact is that these two organizations are running large backbone networks with numerous distributed intrusion detection systems. Given the number of network scanning alerts compared to other types of alerts, e.g., brute-forcing, it is no surprise that the rules containing network scanning alerts are in the Top 10. Since the sequential rule mining does not work with time, the minimal and average time differences between the rules are taken from events matching the rules. This illustrates how much time would be left to mitigate a predicted attack.

Closer inspection of the mined rules shows common situations reflected in the rules. First, we would like to pinpoint rules 7 and 9. In these rules, we can see a network scanning event reported by two intrusion detection systems from two separate networks of different organizations. Similar situations can be seen in rules 1 and 2, where the same network event is detected and reported by different intrusion detection systems from the same organization's network. However, in such cases, one intrusion detection system is typically host-based, while the other is network-based. Thus, the first intrusion detection system observes events related to just one target IP address and the second oversees the whole IP address space, including the IP address of the host-based system. In such cases, we can receive a report of a single attack from multiple intrusion detection systems. Such alerts should be a subject of aggregation to avoid influencing the analysis [28] unless the intrusion detection systems complement each other. For example, in rule 1, one intrusion detection system reports network scanning without specifying the port, while the second intrusion detection system adds information about the ports.

Other interesting observations are related to sequences of actions performed by the attackers. For example, in rule 4, we can see the implications that if an attacker scans the networks on port 2323, the scan of port 23 will follow in order of minutes. Similar port combinations in Top-10 rules can be seen in rules 5 and 8. In general, we can see common combinations of port numbers that are often scanned together [23]. Another example of attack progression can be seen in rule 2, where we can see two distinct attack steps, scanning and brute-forcing. Scanning would intuitively take place before brute-force, but

TABLE II
SUPPORT AND CONFIDENCE VALUES OF TOP-10 SEQUENTIAL RULES DURING THE EXPERIMENT

Rule	**Day 1** (133,785 seq.)		**Day 2** (129,180 seq.)		**Day 3** (137,364 seq.)		**Day 4** (140,093 seq.)		**Day 5** (140,844 seq.)	
	Support	**Confidence**	**Support**	**Confidence**	**Support**	**Confidence**	**Support**	**Confidence**	**Support**	**Confidence**
1	0.00438	0.88386	0.00544	0.89453	0.00468	0.86909	0.00595	0.90554	0.00580	0.90476
2	0.00824	0.53465	0.00955	0.54844	0.00750	0.57953	0.00733	0.59387	0.00655	0.56178
3	0.01987	0.68214	0.02789	0.76877	0.02637	0.77863	0.02558	0.74947	0.02641	0.74415
4	0.06655	0.70099	0.06864	0.71114	0.06246	0.71855	0.06838	0.74378	0.06551	0.75104
5	0.00834	0.58155	0.00818	0.58045	0.00708	0.59474	0.00758	0.55777	0.00930	0.58606
6	0.00487	0.89071	0.00557	0.87378	0.00537	0.86925	0.00727	0.89938	0.00739	0.89356
7	0.00544	0.80088	0.00587	0.89504	0.00546	0.89618	0.00524	0.88341	0.00559	0.89545
8	0.00289	0.9	0.00129	0.78403	0.00138	0.86758	0.00119	0.59011	0.00130	0.77542
9	0.00411	0.60284	0.00414	0.62941	0.00397	0.64311	0.00369	0.60023	0.00401	0.62431
10	0.00266	0.83962	0.00412	0.87070	0.00355	0.83022	0.00478	0.88859	0.00427	0.875

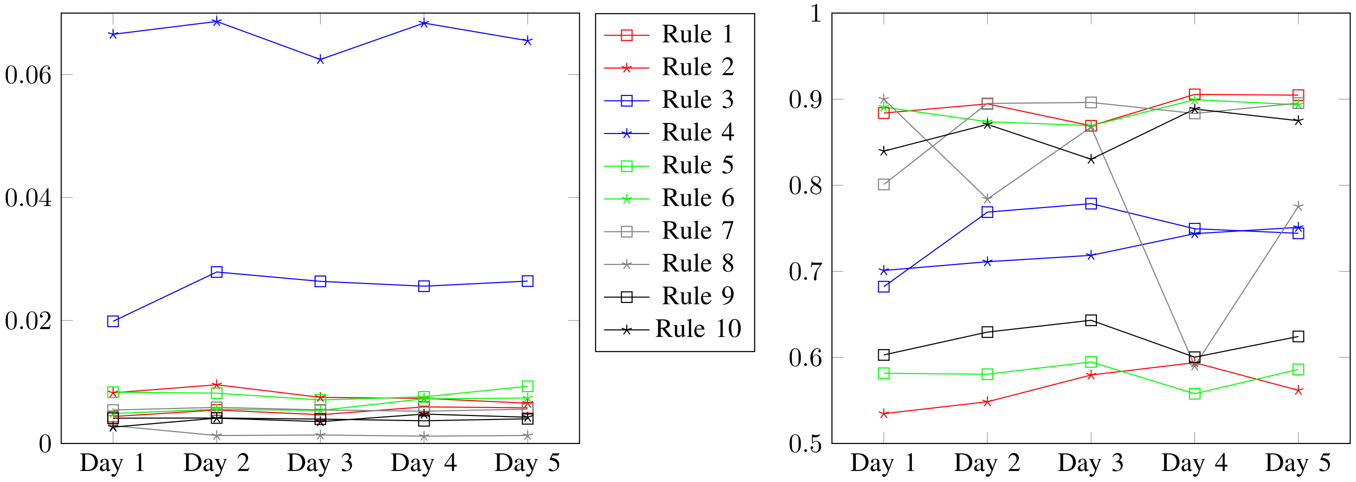

Fig. 1. Evolution of support (left) and confidence (right) values in sequential rules in consecutive day.

in the situation reflected by the rule, we see brute-forcing reported by a network-based intrusion detection system and scanning reported by a honeypot. Thus, the order can appear counterintuitive but makes sense with the knowledge of the deployment details. This only underlines the practical implications of different deployment strategies and architectures of distributed intrusion detection systems [2].

Table II shows support and confidence values of the rules mined during the experiment. From the left, there are the Top-10 rules mined on the first day with their support and confidence values. The remaining columns show support and confidence values of the same rules in the consecutive days, under the condition that the rules were mined again on these days. The number of sequences, from which the rules were mined, is displayed in parentheses next to captions of the days. The gray background in Table II indicates rules that previously appeared in Top-10 rules, but were not present in Top-10 rules at the day of mining.

To better reflect the evolution of support and confidence values over time, we plotted two graphs as can be seen in Fig. 1. We can see that support values were stable for all rules during the experiment. The confidence values are stable as well, although there is an exception in rule 8, which fluctuates a lot. However, we consider rule 8 as an anomaly, as it was observed on the first day of the experiment in Top-10 rules, but it did not appear in Top-10 rules any other day. Nevertheless, the results suggest that most of the rules are applicable for predicting the attacks at least during the following days after they are mined.

VII. CONCLUSION

In this paper, we have empirically evaluated the possibilities of cyber attack prediction based on information exchange and supported by data mining. We used real-world security alerts shared within SABU, an alert sharing platform to run our experiment. Data mining methods, namely sequential rule mining, were used to infer prediction rules for later use. We discussed our empirical experience in obtaining and filtering the predictive rules for their practical usability and evaluated the rules in an experiment. We found out that it is possible to predict the continuation of an attack, i.e., the following event caused by an attacker and the assumed target. Numerous attack patterns are recurring, and their continuations can be predicted with high accuracy, although a significant amount of prediction rules were found to be unfit for practical use. Thus, proper filtering and even manual inspection of the results of data mining should be considered for deployment. On the other hand, if a prediction rule is inferred one day, it is very likely to be applicable in the consecutive days with similar parameters. Thus, predictions based on a certain rule are likely to succeed at least during the following days or weeks.

We are going to further develop the sharing platform SABU and include the prediction framework as one of its components. Based on our empirical experience, the prediction framework will have to include tools for automated filtering and updating of the sequential rules, as well as a user interface for their manual inspection and filtering. Proper engineering of the framework, visualization of the attack patterns, and further research into attack prediction are left for future work. We also hope in encouraging other researchers and developers into creating similar frameworks in other collaborative platforms, as the capability of predicting the next target of an attack in a larger scope illustrates the possible options of attack predictions in collaborative environment [3].

Acknowledgment

The authors would like to express their sincere gratitude to the anonymous reviewers for their constructive feedback. This research was supported by the Security Research Programme of the Czech Republic 2015 - 2020 (BV III / 1 VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20162019029 The Sharing and analysis of security events in the Czech Republic.

References

[1] C. V. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," *Computers & Security*, vol. 29, no. 1, pp. 124 – 140, 2010.

[2] C. Fung and R. Boutaba, *Intrusion Detection Networks: A Key to Collaborative Security*. CRC Press, 2013.

[3] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Comput. Surv.*, vol. 47, no. 4, pp. 55:1–55:33, May 2015.

[4] ENISA, "Detect, SHARE, Protect – Solutions for Improving Threat Data Exchange among CERTs," https://www.enisa.europa.eu/activities/cert/support/information-sharing/detect-share-protect-solutions-for-improving-threat-data-exchange-among-certs/at_download/fullReport, Oct. 2013.

[5] ——, "Standards and tools for exchange and processing of actionable information," https://www.enisa.europa.eu/activities/cert/support/actionable-information/standards-and-tools-for-exchange-and-processing-of-actionable-information, Nov. 2014.

[6] X. Qin and W. Lee, "Attack plan recognition and prediction using causal networks," in *Computer Security Applications Conference, 2004. 20th Annual*, Dec 2004, pp. 370–379.

[7] CESNET and Masaryk University, "SABU," https://sabu.cesnet.cz/en/start, Jul. 2016.

[8] J. Steinberger, A. Sperotto, M. Golling, and H. Baier, "How to exchange security events? Overview and evaluation of formats and protocols," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 261–269.

[9] H. Debar and A. Wespi, "Aggregation and correlation of intrusion-detection alerts," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2001, pp. 85–103.

[10] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer, "Comprehensive approach to intrusion detection alert correlation," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 3, pp. 146–169, July 2004.

[11] F. Cuppens and A. Miege, "Alert correlation in a cooperative intrusion detection framework," in *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, 2002, pp. 202–215.

[12] L. Wang, A. Liu, and S. Jajodia, "Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts," *Computer communications*, vol. 29, no. 15, pp. 2917–2933, 2006.

[13] J. Lei and Z. Li, "Using network attack graph to predict the future attacks," in *Communications and Networking in China, 2007. CHINACOM '07. Second International Conference on*, Aug 2007, pp. 403–407.

[14] Z. t. Li, J. Lei, L. Wang, and D. Li, "A data mining approach to generating network attack graph for intrusion prediction," in *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, vol. 4, Aug 2007, pp. 307–311.

[15] D. Ourston, S. Matzner, W. Stump, and B. Hopkins, "Applications of hidden Markov models to detecting multi-stage network attacks," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, Jan 2003.

[16] H. Farhadi, M. AmirHaeri, and M. Khansari, "Alert Correlation and Prediction Using Data Mining and HMM," *ISeCure*, vol. 3, no. 2, 2011.

[17] Y.-H. Kim and W. H. Park, "A study on cyber threat prediction based on intrusion detection event for apt attack detection," *Multimedia Tools and Applications*, vol. 71, no. 2, pp. 685–698, 2014.

[18] C.-B. Jiang, I.-H. Liu, Y.-N. Chung, and J.-S. Li, "Novel intrusion prediction mechanism based on honeypot log similarity," *International Journal of Network Management*, vol. 26, no. 3, pp. 156–175, 2016.

[19] C. Cipriano, A. Zand, A. Houmansadr, C. Kruegel, and G. Vigna, "Nexat: A history-based approach to predict attacker actions," in *Proceedings of the 27th Annual Computer Security Applications Conference*, ser. ACSAC '11, 2011, pp. 383–392.

[20] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li, "AI^2: Training a Big Data Machine to Defend," in *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, April 2016, pp. 49–54.

[21] X. Ma, J. Zhu, Z. Wan, J. Tao, X. Guan, and Q. Zheng, "Honeynet-based collaborative defense using improved highly predictive blacklisting algorithm," in *2010 8th World Congress on Intelligent Control and Automation*, July 2010, pp. 1283–1288.

[22] J. Freudiger, E. De Cristofaro, and A. E. Brito, *Controlled Data Sharing for Collaborative Predictive Blacklisting*. Cham: Springer International Publishing, 2015, pp. 327–349.

[23] M. Husák, J. Kašpar, E. Bou-Harb, and P. Čeleda, "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*, 2017, pp. 22:1–22:10.

[24] C. Fachkha, E. Bou-Harb, A. Boukhtouta, S. Dinh, F. Iqbal, and M. Debbabi, "Investigating the dark cyberspace: Profiling, threat-based analysis and correlation," in *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS)*, Oct 2012.

[25] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 4, pp. 2037–2064, Fourthquarter 2014.

[26] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An Overview of IP Flow-Based Intrusion Detection," *Communications Surveys & Tutorials, IEEE*, vol. 12, no. 3, pp. 343–356, Third 2010.

[27] CESNET, "Warden," https://warden.cesnet.cz/en/index, Jul. 2017.

[28] M. Husák, M. Čermák, M. Laštovička, and J. Vykopal, "Exchanging security events: Which and how many alerts can we aggregate?" in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 604–607.

[29] M. Husák and M. Čermák, "A graph-based representation of relations in network security alert sharing platforms," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 891–892.

[30] CESNET, "Intrusion Detection Extensible Alert," https://idea.cesnet.cz/en/index, Dec. 2016.

[31] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)," RFC 4765 (Experimental), Internet Engineering Task Force, Mar. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4765.txt

[32] P. Fournier-Viger and V. S. Tseng, "Mining top-k sequential rules," in *International Conference on Advanced Data Mining and Applications*. Springer, 2011, pp. 180–194.

[33] P. Fournier-Viger, J. C.-W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, "The SPMF Open-Source Data Mining Library Version 2," in *Proc. 19th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2016)*. Part III, Springer LNCS 9853, 2016, pp. 36–40.