Limiting the Size of a Predictive Blacklist While Maintaining Sufficient Accuracy

Samuel Šuľan
Faculty of Informatics
Masaryk University
Brno, Czech Republic
kalat@mail.muni.cz

Martin Husák
Institute of Computer Science
Masaryk University
Brno, Czech Republic
husakm@ics.muni.cz

ABSTRACT

Blacklists (blocklists, denylists) of network entities (e.g., IP addresses, domain names) are popular approaches to preventing cyber attacks. However, the limited capacity of active network defense devices may not hold all the entries on a blacklist. In this paper, we evaluated two strategies to limit the size of a blacklist and their impact on the blacklist's accuracy. The first strategy is setting the maximal size of a blacklist; the second is setting an expiration time to blacklist items. Short-term attack predictions are typically more precise, and, thus, the recent blacklist entries should be more valuable than older ones. Our experiment shows that the blacklists reduced to half of the size via either strategy achieve only a 25 % drop in accuracy.

CCS CONCEPTS

• **Security and privacy → Network security**; • **Applied computing → Operations research**.

KEYWORDS

cybersecurity, blacklist, limitation, prediction

ACM Reference Format:
Samuel Šuľan and Martin Husák. 2022. Limiting the Size of a Predictive Blacklist While Maintaining Sufficient Accuracy. In *The 17th International Conference on Availability, Reliability and Security (ARES 2022), August 23–26, 2022, Vienna, Austria.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3538969.3539007

1 INTRODUCTION

The rising numbers and severity of cybersecurity incidents call for the application of novel approaches to network security management, both in incident response but also in prevention. One of the most popular preventive approaches is employing blacklists (blocklists, denylists). A blacklist contains a list of network entities (e.g., IP addresses, domain names) that were or are supposed to behave maliciously. The network administrators use the blacklists to deny access to their networks, systems, and service to the network entities on the blacklist. A blacklist can be created using the observations of malicious activities in one's own network or can be outsourced. Various cyber threat intelligence (CTI) feeds [16], distributed intrusion detection systems [15], and alert sharing platforms [8] can be used to obtain a blacklist or raw data for creating one.

One of the major issues of blacklisting is the relevance of the entries. Putting any maliciously-behaving IP address on a blacklist is simple, but there is no guarantee that the malicious activity will continue or repeat. A blacklist based on observations collected in one environment might not be relevant for preventing malicious activities in a different environment. These factors result in the low applicability of most of the blacklist content. Researchers and practitioners are trying to filter the data to increase the relevancy of the blacklist. For example, CTI feeds are often focused on particular environments so that the users may pick the most fitting blacklist for their geographical location, organization type, or threatened services and devices. An advanced approach is using predictive blacklists that are based on predictions of network attacks that are already aimed specifically and promise higher relevance. We typically refer to such attempts as predictive or personalized blacklisting [5].

Another issue of blacklisting is the size of the blacklist. Naturally, the list of network entities is significantly smaller than the raw data from which it was inferred, such as intrusion detection alerts, network traffic records, or system logs. Still, the number of IP addresses and other network entities may be too large to employ them. The blacklists are typically converted into firewall or routing rules, and the capacity of the active network devices is limited. There is a need to limit the size of the blacklist in order to fit the capacity of the active network devices.

To approach the highlighted issues, we build upon previous work on predictive blacklisting and study the effects of two blacklist size limitation strategies. We follow previous work on the design of an attack prediction system [6] that is used to generate predictive blacklists [5]. With the help of real-world attack samples [8], we generate various blacklists limited in size and compare them to the unlimited blacklist. First, we set various maximal sizes of the blacklist and use only the most N recent entries. Second, we set an expiration time for each blacklist item. In an experiment, we aim to answer the following research questions:

- How can we cut a predictive blacklist in size while maintaining sufficient accuracy?
- After what time could the entries in the predictive blacklist expire?
- Which of the two strategies provides better results in terms of blacklist size and the number of changes?

This paper is structured into 6 section. In Section 2, we summarize the background and related work. The experiment setup is

ARES 2022, August 23–26, 2022, Vienna, Austria

presented in Section 3 and the results are presented in Section 4. Section 5 concludes the paper.

2 BACKGROUND AND RELATED WORK

Herein, we first describe the background of blacklisting in cybersecurity with a focus on the methods of predictive blacklisting and their evaluation. Second, we briefly survey related work.

2.1 Background

Following the terminology of related work [13, 18], the blacklist may stand in the form of Local Worst Offender List (LWOL) or Global Worst Offender List (GWOL). In the case of LWOL, the blacklist is generated from the internal records of an organization, system, or service and contains, for example, the most frequent network entities that behaved maliciously. Such network entities are recognized by local Intrusion Detection Systems (IDS) or similar means. The advantage of LWOL is that local administrators have full control over it and its creation. However, such a blacklist is often purely reactive and prevents only repeating attacks. It does not protect against threats not observed in the local environment and does not act proactively.

On the contrary, GWOL is a global list, mostly collected by a third party using data from IDS, honeypots, and other sources distributed globally. A well-known example of a GWOL source is DShield[1], which processes around 30 million alerts per day. Such an amount of input data generates a huge blacklist of a plethora of attackers observed globally. When using such a blacklist locally, many attacks not observed locally could be prevented. However, the blacklists are impractically large, and it is very likely that most of their content is not relevant to the local environment.

When evaluating blacklist, we typically use two metrics, *hit count* and *hit rate*. The hit count is the number of attacks prevented by deploying a blacklist. The hit rate is the hit count divided by the size of the blacklist, usually measured in the number of entries.

Predictive blacklisting is a proactive variation of blacklisting. The entries in a predictive blacklist were not necessarily observed in the past by IDS or a similar tool but, instead, they are expected to behave maliciously in the future. The advantage of such a blacklist is that it is more likely to achieve a higher hit rate and, thus, protect the infrastructure more efficiently. The predictive blacklists are commonly based on attack prediction or attack projection [7].

Attack projection refers to a number of techniques used to project an ongoing cyber attack. For example, if there is a known attack scenario and we observe an attacker who already performed a part of the sequence, we may project the attack and predict that the attacker will perform the remainder of the steps in the sequence. In the past, the attack projection used predefined attack libraries that were impractical to use, while modern approaches use data mining and machine learning to project the attacks autonomously. An example of such an approach is the work by Husák et al. [5] implemented as the AIDA framework [6]. Attack projection is effective for making predictions about the near future and is backed by recent observations of potentially malicious behavior [7]. Another advantage is that the prediction also states what kind of attack is expected to be performed and against which target, which could be further exploited by the defenders.

Attack prediction is a more generic approach that can use other techniques to predict an attack or designate a network entity as potentially malicious. Contrary to attack projection, there is no need to have recent evidence of potentially malicious activity of a network entity. For example, the approaches based on reputation scoring use long-term observation, contextual information, and even profiling to set the reputation of a network entity. The work by Bartoš et al. [2], implemented in the NERD tool [1], uses up to a 1-year-long history of intrusion detection alerts, honeypot logs, presence of an entity on other blacklists, and other data sources to generate so-called Future Misbehavior Probability (FMP) score. The higher the FMP score is, the more likely it is that the network entity will behave maliciously in the future. This approach is effective in protecting against frequent abusers and long-lasting threats. However, we may not know what kind of malicious activity will the network entity perform in the future or where that will take place.

2.2 Related Work

The first mentions of predictive blacklisting could be traced back to the work of Zhang et al. [17, 18]. The authors illustrate how a predictive blacklist achieves higher accuracy than simple blacklists based on LWOL and GWOL. The authors' approach is inspired by Google's PageRank algorithm. Soldo et al. [13, 14] used three approaches to predict attacks and generate predictive blacklists. The authors used time series-based predictions, adaptation of k-NN clustering to reflect the similarity between the targets of the same attackers, and a co-clustering algorithm that discovers the groups of attackers attacking the same target at the same time. Ma et al. [10] focused on the data from honeypots and creating personalized blacklists. However, the accuracy of their approach is lower.

While the earlier approaches typically used all the data they had available, Freudiger et al. [4] focused on the shared data from which the blacklists are generated in a collaborative environment. The authors show that sharing only the data on common attacks is almost as effective as sharing all data. Melis et al. [11] compared the collaborative predictive blacklist in which the trusted peers see all the data [13] to a privacy-preserving data sharing [4]. Their main finding is that collaboration increases both the number of predicted attacks and the false positives. Therefore, they present a hybrid approach with better trade-offs of true and false positives.

Jeong and Tak [9] applied machine learning to the predictive IP address blacklist. As they state, we are yet to see a mature ML-based solution, but their approach mitigates open challenges and provides a nearly 90 % reduction of incorrect blacklisting compared to the performance of human experts.

The recent implementations of the attack prediction and projection systems capable of generating a predictive blacklist, such as the NERD [1, 2] and AIDA framework [5, 6], were described in the previous subsection. Their detailed comparison, including the discussion of their usability, was recently published by Husák et al. [7].

[1] https://www.dshield.org/

To illustrate that IP addresses are not the only network entities considered for a predictive blacklist, we would like to highlight two representative works using other entities and use cases. Felegyhazy et al. [3] discussed proactive domain blacklisting. The authors show how to predict the maliciousness of a domain name by leveraging its properties inherent to its registration and appearance in DNS zone files. In another example, Prakash et al. [12] explored the use of predictive blacklisting and its application to phishing detection. The combinations of components of known malicious URLs are used to infer new phishing URLs.

3 EXPERIMENT SETUP

In this section, we describe the experiment setup. First, we briefly present the AIDA framework and the dataset that we used to generate the blacklists. Subsequently, we describe our approach to blacklist analysis and the implementation of the two strategies to limit the size of the blacklist. The results are summarized in the following section.

3.1 Data and tools

In the experiment, we used the dataset [8] of intrusion detection alerts collected from the SABU alert sharing platform. The same dataset was used in previous works [5, 7], which shall allow for comparison. The data was collected continuously as they appeared in the SABU platform for the period of one week, from March 11 to March 17, 2019. Almost 12 million alerts were collected from 34 network-based IDS, honeypots, and other data sources deployed in 3 distinct organizations: national research and education network, a large campus network, and a commercial Internet service provider. The alerts are stored in the IDEA format and categorized using the taxonomy of security events included in the IDEA definition. The IP addresses in the alerts are anonymized.

The data were processed using the AIDA framework [6]. AIDA is a modular framework for the stream-based analysis of intrusion detection alerts using the concepts of big data processing, data mining, and complex event processing. The framework receives intrusion detection alerts and distributes them to several components that perform the tasks of data sanitization, mining predictive rules, making predictions, and calculating the accuracy of the predictions. The framework uses Top-k sequential rule mining to infer predictive rules. A predictive rule consists of two ordered sets of actions (e.g., network scanning, brute-force password attack, exploitation); the first rule implies the other. For example, if a rule states $A, B => C$, then if there is an IP address that was observed to perform actions A and B, it is predicted that this IP address will also perform action C. All the IP addresses that are predicted to perform such actions are put on a predictive blacklist.

The AIDA framework was then deployed in a virtual machine running Ubuntu 18.04 operating system and equipped with a 20 GB hard drive, 6 GB RAM, and 2 CPU cores. Otherwise, the default configuration of the AIDA framework was used. Following the experiment setup of previous works [5, 7], we split the dataset into seven parts, each corresponding to one day of malicious activities. For each day, we used the AIDA framework's data mining component to infer the set of attack prediction rules. The rules from one day are then used to predict an attack on a consecutive day, which is inspired by the real-world deployment of the AIDA framework [6]. For example, the rules inferred from Monday's data are used to predict the attacks in Tuesday's data.

There is an issue with processing datasets with the AIDA framework. In real-time data processing, the current time is used as a timestamp of a prediction. When processing a dataset of older alerts, the current time would not make much sense. Thus, the latest timestamp of an event on which the prediction is made is used as the time of the prediction. For example, when having a rule $A, B => C$, the timestamp of event B is used as a time when the prediction of C was made. The time spent on forwarding the events from an IDS to the AIDA framework and processing them in the framework is neglected in this case.

3.2 Implementation details

In order to evaluate the effectiveness of blacklist entries, we defined the 3-tuple $(startTime, endTime, blockedAlerts)$ describing the presence of an IP address on a blacklist. The entries ($startTime$, $endTime$) show the time when the IP address was added to the blacklist and removed from it. The *BlockedAlerts* is the number of attacks observed within this time interval and originating from the IP address. Such attacks could be prevented by the blacklist and are also referred to as *prevented attacks* in the remainder of this paper. Each IP address may have several 3-tuples assigned to it, namely when it was removed from the blacklist and then inserted again. The 3-tuples are stored in a list representing a blacklist.

The size limitation is implemented as follows. The blacklist of size P is simulated as a priority queue of size P. The predictions are processed sequentially from the oldest to the newest. The IP address and the prediction time (named N, in seconds) are extracted from the prediction. A new 3-tuple is created for the IP address and added to the queue; the priority equals the number of processed predictions, and the 3-tuple contains $(N, N, 0)$. If the IP address is already in the list, then its priority is overwritten. When the queue is full, the IP address with the lowest priority is removed, and the second value of its 3-tuple is set to N. When all the predictions are processed, the second values of all the 3-tuples still on the blacklist are set to the end of the processed day. The experiment was first conducted with the initial blacklist size of 100. In the following iterations, we increased the blacklist size by 500 each iteration until we reached the number of unique IP addresses in the predicted events.

The time expiration is implemented as follows. All unique IP addresses in the predictions are iterated. Subsequently, the list of values of time expiration is iterated, and the 3-tuples are created. The 3-tuples have the form $(N, N+P, 0)$, where N is the time when the predicted event was detected, and P is the expiration time in seconds. If the processed prediction has the detection time N in the interval $< startTime, endTime >$, the $endTime$ is set to $N + P$. The experiment was rerun several times with different expiration settings. We started with the value of 1,800 seconds (0.5 hour) and then repeated the experiment with the value incremented by 1,800 seconds until we reached 24 hours.

Setting the final value of $blockedAlerts$ is implemented in the same way for both strategies. All the alerts were iterated, and if they describe an attack originating from an IP address for which

a 3-tuple is defined, the timestamp of the alert is checked against the time interval in the 3-tuple. If the timestamp is within the time interval of the 3-tuple, the *blockedAlerts* value is incremented by 1.

3.3 Watched metrics

The metrics watched throughout the experiment are as follows: median and average mitigation time, percentage of prevented attacks, number of changes to the blacklist, and the maximal and average size of the blacklist.

The mitigation time is the amount of time between the prediction of an event by the AIDA framework and the subsequent detection of this event in case it actually happens. This metric is highly valuable in practice because a longer mitigation time gives the network operators more time to react to the prediction, e.g., by putting in on a blacklist and updating a firewall rule.

The percentage of prevented attacks is the number of observed attacks originating from the IP addresses on the blacklist divided by the total number of observed attacks. This metric shows how many attacks could have been prevented if the blacklist had been used.

The number of changes to the dataset is the number of operations of inserting or deleting an IP address to or from a blacklist. Replacing one IP address with another is considered two changes. The reason we watch this metric is that the frequency of changes might be more important than the size of the blacklist used by an active network defense device. In practice, it may take even several minutes for a new blacklist entry to propagate into a firewall or routing rule and become efficient, namely at high-throughput network devices.

The maximal size of a blacklist is the highest number of IP addresses on a blacklist over time. The average size of a blacklist in one hour is calculated as:

$$\sum_{n=f}^{f+3600} Blacklist_f/3600 \tag{1}$$

where f is the beginning of the considered hour. $Blacklist_f$ is the size of the blacklist at f. The number of changes and the maximal and average size of a blacklist are relevant for practical reasons, namely the capacity of active network defense devices.

4 EXPERIMENT RESULTS

In this section, we present the experiment results. First, we show the overview of the raw data, predictions, and unique IP addresses. Subsequently, we show the overall blacklist accuracy and compare the representative blacklists. Table 1 shows the detailed statistics of the dataset and predicted events per day. For each day in the dataset, we display the number of events on which we predicted the attacks and the number of events with a predicted IP address. Subsequently, we display the number of predictions and the number of unique IP addresses in the predicted event. As we can see, many predictions are related to the same IP address. Finally, the average and median mitigation times of the predictions are displayed.

The first results are presented in two Figures illustrating the effects of applying the two strategies to blacklist size reduction. For recapitulation, the key metric is the percentage of prevented attacks, i.e., the percentage of attacks that could be prevented by restricting the network traffic of the IP addresses on the blacklist.

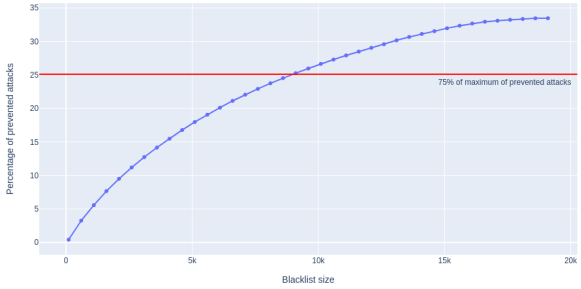


Figure 1: Blacklist accuracy with various size limitations.

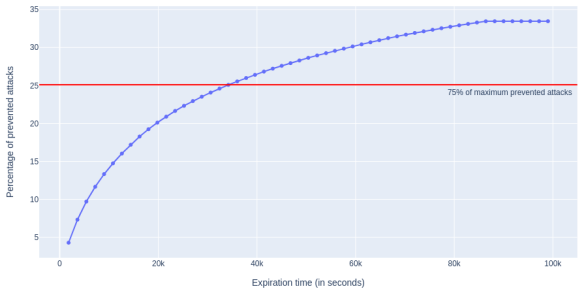


Figure 2: Blacklist accuracy with various item expiration times.

For comparison, the unlimited blacklist is capable of preventing 33.46 % of all the attacks, including the attacks from IP addresses that were not predicted to behave maliciously. When considering only the attacks from the predicted IP addresses, the unlimited blacklist is capable of preventing 88.09 % of such attacks. In the remainder of this section, we use the first metric.

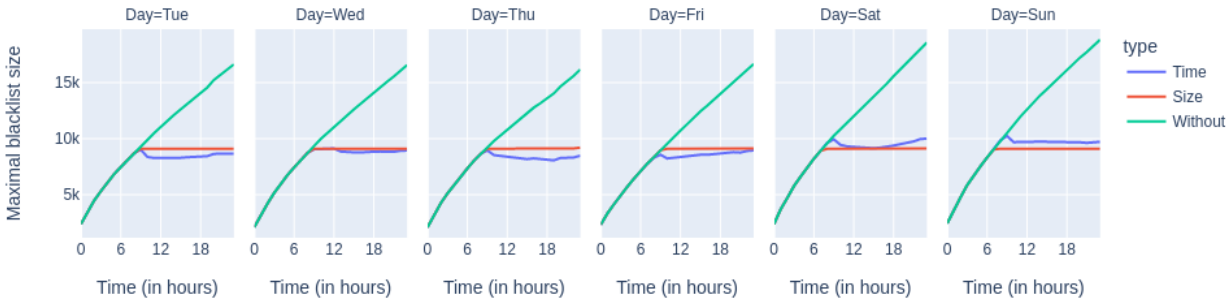
Figure 1 shows the size of the blacklist on the horizontal axis and the percentage of prevented attacks on the vertical axis. We could see what would be the effect of using the blacklist if the limitation of its size were applied. The red horizontal line displays the level at which the size-limited blacklist would achieve 75 % accuracy compared to the full-size blacklist. As we can see, such accuracy would be achieved with a blacklist limited to approximately 9,100 IP addresses. Such a blacklist prevented 25.26 % of the attacks and was selected as the first representative blacklist for further evaluation.

Figure 2 illustrates the effect of the second strategy, the blacklist item expiration. The vertical axis again shows the percentage of prevented attacks. The horizontal axis shows the blacklisted item expiration time in seconds. We again highlighted a level of 75 % blacklist accuracy. As we can see, such a level would be achieved with an expiration time set to around 10 hours. Such a blacklist prevented 25.54 % of the attacks and was selected as the second representative blacklist.

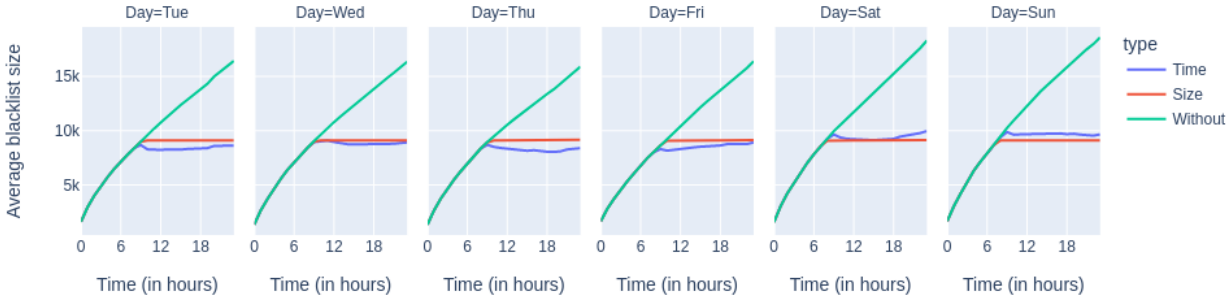
The detailed breakdown of the experiment results per day is presented in Figure 3. Figures 3a and 3b show the maximal and

Day	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Total
Number of events	1,664,838	1,548,300	1,607,747	1,710,206	1,776,281	1,699,196	10,006,568
Number of events with a predicted IP address	645,123	577,048	577,787	644,360	672,507	672,992	3,789,817
Number of predictions	45,046	46,992	39,240	39,410	42,418	41,451	254,557
Number of unique IP addresses in prediction	16,644	16,583	16,090	16,643	18,559	18,854	64,021
Average mitigation time of the predictions (in minutes)	65.59	54.90	62.91	61.07	65.22	69.45	64.90
Median mitigation time of the predictions (in minutes)	26.02	16.81	22.60	22.35	22.50	25.60	23.80

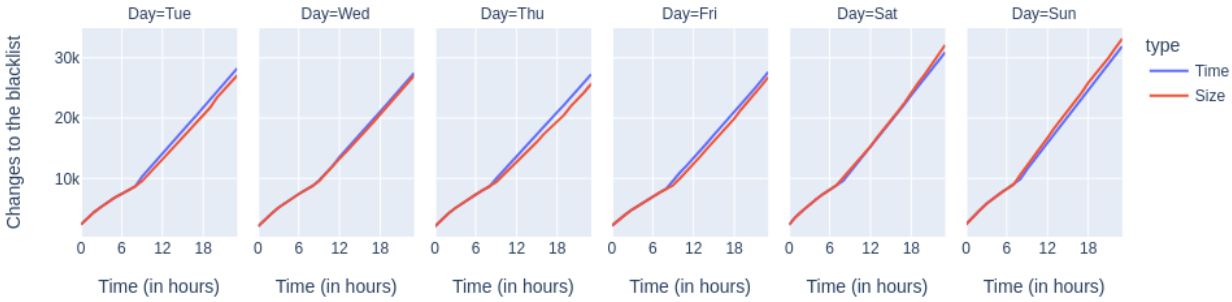
Table 1: Daily statistics of the experiment data.



(a) Maximal size of representative blacklists.



(b) The average size of representative blacklists.



(c) The number of changes to the blacklists in time.

Figure 3: The maximal and average sizes and the number of changes to representative blacklists each day.

Day	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Total
Time expiration	25.91	26.88	23.77	24.60	25.58	25.81	25.54
Size limit	26.73	27.08	24.16	25.55	24.99	24.83	25.26
Unlimited blacklist	34.83	31.84	30.51	33.47	33.10	35.28	33.46

Table 2: Attacks prevented by representative blacklists (in percentages).

average sizes of the representative blacklists and their comparison to unlimited blacklists. As we can see, the unlimited blacklists grow in size continuously, while the limited blacklists reach their capacity after around 10 hours and remain at it in case of size-limited blacklists or oscillate around it in case of timed expiration of entries. It is worth noting that the maximal size of the size-limited blacklist is 49.46 % of the size of the unlimited blacklist, while the maximal size of the blacklist with time expiration is 56.73 % of the unlimited blacklist. Figure 3c shows the frequency of changes. As we can see, the blacklists are similar in this regard. When compared to Figure 3b, we may notice that the shorter the average size of the blacklist is, the more changes it requires. Obviously, this is caused by the frequent deletion of entries that are no longer necessary. Finally, Table 2 shows the percentage of attacks prevented each day.

5 CONCLUSION

The goal of this work was to find out how we can limit the number of entities (e.g., IP addresses) in a predictive blacklist while maintaining sufficient accuracy of the blacklist. We conducted an experiment in which we used the AIDA framework, an attack prediction framework, to generate a predictive blacklist out of a real-world dataset obtained from an alert sharing platform. Subsequently, we applied two strategies to limit the size of the blacklist with various parameters. In particular, we set the maximal size of the blacklist and implemented the expiration of blacklist entries.

The experiment results suggest that a blacklist reduced in size may still display sufficient accuracy when compared to the use of the full blacklist, namely when the most recent entries are used, regardless of the strategy. A blacklist reduced to half of the size may have an accuracy of around 75 % of the original blacklist. Nevertheless, the final selection on which strategy to use lies on the network operators and the available equipment. The important metric to follow would be the number of changes to the blacklist. If the change to the blacklist is a cheap operation, then the size limitation of the blacklist seems to be more suitable.

In our future work, we would like to repeat the experiment in a longer continuous time window and also in a real-world scenario using active network defense devices with limited capacity. Thus, we could observe delays caused by forwarding the blacklists, reconfiguration of the device, and other practical issues. We believe our work will help in the development of more precise active network defense mechanisms that could make use of distributed equipment.

ACKNOWLEDGMENTS

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).

REFERENCES

[1] Václav Bartoš. 2019. NERD: Network Entity Reputation Database. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*. ACM, New York, NY, USA, Article 84, 7 pages.
[2] 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.
[3] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. 2010. On the Potential of Proactive Domain Blacklisting. In *3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET 10)*. USENIX Association, San Jose, CA, 6 pages.
[4] Julien Freudiger, Emiliano De Cristofaro, and Alejandro E. Brito. 2015. *Controlled Data Sharing for Collaborative Predictive Blacklisting*. Springer International Publishing, Cham, 327–349.
[5] 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, Article 19 (Sep 2020), 16 pages.
[6] 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*. ACM, New York, NY, USA, Article 81, 8 pages.
[7] 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.
[8] Martin Husák, Martin Žádník, Václav Bartoš, and Pavol Sokol. 2020. Dataset of intrusion detection alerts from a sharing platform. *Data in Brief* 33 (2020), 106530.
[9] Dooyong Jeon and Byungchul Tak. 2019. BlackEye: automatic IP blacklisting using machine learning from security logs. *Wireless Networks* 28 (2019), 937–948.
[10] Xiaobo Ma, Jiahong Zhu, Zhiyu Wan, Jing Tao, Xiaohong Guan, and Qinghua Zheng. 2010. Honeynet-based collaborative defense using improved highly predictive blacklisting algorithm. In *2010 8th World Congress on Intelligent Control and Automation*. IEEE, New York, NY, USA, 1283–1288.
[11] Luca Melis, Apostolos Pyrgelis, and Emiliano De Cristofaro. 2019. On collaborative predictive blacklisting. *Computer Communication Review* 48, 5 (2019), 9–20.
[12] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, and Minaxi Gupta. 2010. PhishNet: Predictive Blacklisting to Detect Phishing Attacks. In *2010 Proceedings IEEE INFOCOM*. IEEE, New York, NY, USA, 5 pages.
[13] Fabio Soldo, Anh Le, and Athina Markopoulou. 2010. Predictive blacklisting as an implicit recommendation system. In *2010 Proceedings IEEE INFOCOM*. IEEE, New York, NY, USA, 9 pages.
[14] Fabio Soldo, Anh Le, and Athina Markopoulou. 2011. Blacklisting Recommendation System: Using Spatio-Temporal Patterns to Predict Future Attacks. *IEEE Journal on Selected Areas in Communications* 29, 7 (2011), 1423–1437.
[15] Emmanouil Vasilomanolakis, Shankar Karuppayah, Max Mühlhäuser, and Mathias Fischer. 2015. Taxonomy and Survey of Collaborative Intrusion Detection. *Comput. Surveys* 47, 4, Article 55 (May 2015), 33 pages.
[16] Thomas D. Wagner, Khaled Mahbub, Esther Palomar, and Ali E. Abdallah. 2019. Cyber threat intelligence sharing: Survey and research directions. *Computers & Security* 87 (2019), 101589.
[17] Jian Zhang, Phillip Porras, and Johannes Ullrich. 2007. A New Service for Increasing the Effectiveness of Network Address Blacklists. In *3rd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI 07)*. USENIX Association, Santa Clara, CA, USA, 6 pages.
[18] Jian Zhang, Phillip Porras, and Johannes Ullrich. 2008. Highly Predictive Blacklisting. In *17th USENIX Security Symposium (USENIX Security 08)*. USENIX Association, San Jose, CA, USA, 107–122.