DomainPrio

Prioritizing Domain Name Investigations to Improve SOC Efficiency

Chiba, Daiki; Akiyama, Mitsuaki; Otsuki, Yuto; Hada, Hiroki; Yagi, Takeshi; Fiebig, Tobias; Van Eeten, Michel

DOI
10.1109/ACCESS.2022.3161636

Publication date
2022

Document Version
Final published version

Published in
IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 12, 2022, accepted March 16, 2022, date of publication March 23, 2022, date of current version April 1, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3161636

DomainPrio: Prioritizing Domain Name Investigations to Improve SOC Efficiency

DAIKI CHIBA[1], (Member, IEEE), MITSUAKI AKIYAMA[2], (Member, IEEE), YUTO OTSUKI[1], HIROKI HADA[1], TAKESHI YAGI[2], TOBIAS FIEBIG[3], AND MICHEL VAN EETEN[4]

[1]NTT Security (Japan) KK, Chiyoda-ku, Tokyo 101-0021, Japan
[2]NTT, Musashino-shi, Tokyo 180-8585, Japan
[3]Max-Planck-Institut für Informatik, Saarbrücken, 66123 Saarland, Germany
[4]Faculty of Technology, Policy and Management, TU Delft, Delft, 2628 BX Zuid-Holland, The Netherlands

Corresponding author: Daiki Chiba (daiki.chiba@ieee.org)

ABSTRACT Security Operations Centers (SOCs) are in need of automation for triaging alerts. Current approaches focus on analyzing and enriching individual alerts. We take a different approach and analyze the population of alerts. In an observational study over 24 weeks, we find a surprising pattern: some domains get analyzed again and again by different analysts, without coming to a final evaluation. Overall, 19% of the domains trigger 74% of all investigations. The most time-consuming domains are classified as false positives and ''potentially unwanted programs''—the lowest threat level. To increase SOC efficiency, we design DomainPrio, a tool that prioritizes domains that are likely to be the subject of repeated, incomplete investigations. This enables us to indicate to the first analyst encountering this domain that the investigation should be, if possible, completed on this first attempt, so future investigations on the same domain can be prevented. DomainPrio is able to predict these domains with 89% accuracy and does so with an interpretable and auditable logistic regression model. When evaluating our tool on 100 days of data from a production setting, we find that it can potentially reduce the number of alert investigations by up to 35%, presenting the SOC with very substantial efficiency gains.

INDEX TERMS Network security, security operations, security operations centers, SOC, threat analysis.

LIST OF ABBREVIATIONS

SOC	Security Operations Center.
MSSP	Managed Security Service Provider.
SIEM	Security Information and Event Management.
IPS	Intrusion Prevention System.
UTM	Unified Threat Management.
SOAR	Security Orchestration, Automation, and Response.
DNS	Domain Name System.
PUP	Potentially Unwanted Program.
FP	False Positive.
C&C	Command and Control.
IQR	Interquartile Range.
TTL	Time-To-Live.
SD	Standard Deviation.
Var	Variance.
TPR	True Positive Rate.
TNR	True Negative Rate.
BAC	Balanced Accuracy.
ISP	Internet Service Provider.

The associate editor coordinating the review of this manuscript and approving it for publication was Congduan Li.

I. INTRODUCTION

Many organizations defend themselves through a patchwork of security controls and threat detection systems. These systems generate a voluminous stream of events and alerts that need to be monitored, analyzed, and acted upon, typically by staff in a Security Operations Center (SOC). For years, academic research and industry reports alike have raised the problem of alert fatigue and analyst burnout [15], [30]. The key response to this problem has been to pursue better automation. Indeed, an ''insufficient automation level of SOC components'' was considered the top issue by SOC managers, according to a recent study [17].

The promise of automation for SOCs is clear: there is a deluge of data and there is repetition in how the analysts engage with the data [29], [31], [35]. Some approaches focus on better separating the true positives from the false positives, typically by estimating anomaly or maliciousness scores for each alert [13], [19], [21], [22], [26]. Other approaches support the triage process by extracting and enriching the most relevant information of an alert [8], [39], [40].

What these approaches have in common is that they fundamentally conceptualize the problem as analyzing each alert on its own merits. In this paper, we take a different approach and analyze the problem at the level of the total population of alerts coming into a SOC. During a six-month observational study in a large SOC of a Managed Security Service Provider (MSSP), we find that analysts repeatedly investigate the same domains showing up in different alerts, without coming to a final evaluation on the domain. A final evaluation of a domain would allow the automated handling of future alerts relating to that domain by the Security Information and Event Management (SIEM) system, at least until the domain would have to be re-evaluated. This would save precious analyst time.

The repeated investigations are concentrated in a fraction of the domains. In fact, we observe a power-law distribution: 19% of the domains lead to 74% of all investigations by triggering five or more analyst engagements. In extreme cases, the same domain might be investigated over 500 times in the course of 24 weeks. Even worse, when we analyze a random sample of 400 of these domains, we find that the majority were not part of high-threat alerts. Instead, they mostly relate to false positives (50%) and non-malicious, but unwanted, software (13%). Those two categories trigger a factor of 12 more incomplete investigations than high-threat domains. In short, a small fraction of all domains consumes a disproportionate amount of time from SOC analysts, while not posing the most significant threat. We corroborate these findings in two other large SOCs in different countries.

Based on these observations, we propose a novel and complementary approach to SOC automation aimed at increasing efficiency and reducing alert fatigue. Rather than predicting the true positives or enriching the information of individual alerts, as prior studies aimed to do, we set out to reduce the number of repeated investigations. For this purpose, we design DomainPrio, a tool to predict which domains will trigger repeated incomplete investigations across the total pool of analysts, thus consuming the most SOC time. The tool presents a twofold change in the process: (1) prioritize domains that will end up consuming the most analyst time and (2) ensure that investigations for these domains are completed, so related alerts can be automatically handled by the SIEM. Using a feature set over SOC analysts' engagements across all domains, our classifier is able to predict domains which will be investigated repeatedly with 89% accuracy. With this prioritization, we save 10–341 subsequent investigations for each domain.

DomainPrio does not require analyst time and effort for labeling or support. It can be automatically trained and updated every day on the most recent features. The tool is scalable and can be easily deployed. The underlying model, a logistic regression, is interpretable and auditable. In short, we make these contributions:

- We conduct a six-month observational study in a real-world MSSP SOC and find that a small fraction of the domains consumes the bulk of analyst time while not posing the most significant threat. We confirm this pattern in other SOCs.
- We design DomainPrio and demonstrate that it can predict with high accuracy which domains will end up consuming repeated investigations, without itself requiring analysts' time for labeling or support.
- By evaluating DomainPrio on a real SOC dataset in a production setting, we find that prioritizing domains can save an upper-bound of about 35% of all investigations over a 100-day period. While in practice the gains will be smaller, this still represents a major efficiency improvement for a SOC.

II. SOC BACKGROUND

We first explore the workflow of an example SOC to better understand SOC analysts' needs in terms of tooling. This allows us to account for those needs and focus our improvement attempts on items relevant to SOC analysts.

A. CASE STUDY SOC

A recent survey of security practitioners [33] found that 51% of organizations have decided to outsource some or all of their SOC functions because of the difficulty of staffing analysts to maintain an in-house SOC. MSSPs collect and analyze alerts from customers' networks and then detect threats and initiate the response procedure on behalf of their customers [1], sometimes also called SOC-as-a-Service. We conduct our study in one SOC of an MSSP that operates a total of 10 SOCs in the Americas, Asia-Pacific (APAC), and Europe, Middle East and Africa (EMEA) regions, staffed by 2,000 cybersecurity professionals. Across all SOCS, the MSSP provides security services to more than 10,000 companies in 58 countries. It analyzes over 61 trillion events per year. While there are organizational differences between in-house and outsourced SOCs, their overall workflow is similar [35]. Hence, we expect our findings to also benefit in-house SOCs, if they are sufficiently large to encounter the scale effects we are leveraging in our case study.

B. SOC WORKFLOW

We outline the standard workflow in the case study SOC in Figure 1, which is comparable to what has been reported in other studies, e.g., [13], [17], [22], [31], [39]. The central task of any SOC is real-time monitoring and analysis of security alerts generated by systems in customer environments, such as Intrusion Prevention Systems (IPS), local web proxies, sandboxes analyzing email attachments, and Unified

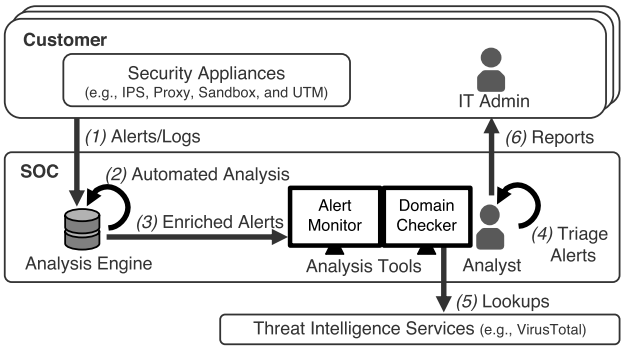

FIGURE 1. SOC workflow overview: *(1)* The SOC receives alerts and tracks them in a SIEM, *(2)* Automated pre-analysis classifies and handles alerts that can be processed with pre-defined rules, *(3)* Alerts that can not be handled automatically are forwarded to analysts, *(4)* An analyst triages the incoming alerts, *(5)* If the triage warrants an investigation, the analyst processes the alert (threat intelligence, domain lookups, etc.), *(6)* The results are reported to the customer.

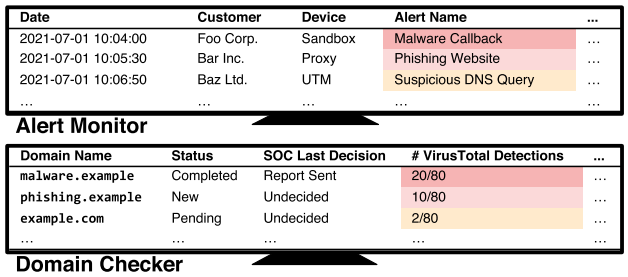

FIGURE 2. Wireframes of the alert monitor and domain checker. These two tools are those in which SOC analysts spend most of their time.

Threat Management (UTM) solutions. Alerts and logs are sent to the SOC's analysis engine, which uses a SIEM to handle the large number—up to 2 trillion alerts and logs per month [26]—of alerts (*(1)* in Figure 1). The analysis engine performs Security Orchestration, Automation, and Response (SOAR), and responds to alerts where there are known malicious characteristics and predefined rules automatically (*(2)* in Figure 1). Alerts that can not be automatically handled are forwarded to the alert monitor for triage by a human analyst (*(3)* in Figure 1), who receive an overview of the alerts (see Figure 2).

An analyst then evaluates the alert based on a further investigation of the information provided in the alert monitor (*(4)* in Figure 1), and lookups of associated domain names, IP addresses, and other indicators in threat intelligence services (*(5)* in Figure 1). In our SOC, analysts can conduct lookups on domains via a specific tool providing threat intelligence results and other meta-information on a domain (Figure 2).

The investigation ends with an evaluation (malicious or benign) and, when needed, in reporting the threat to the customer (*(6)* in Figure 1). Note that there is no unified score indicating whether it is worthwhile to evaluate a domain.

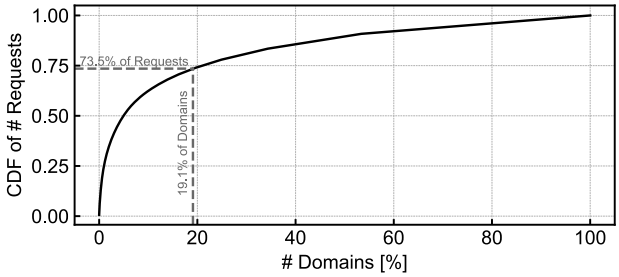

FIGURE 3. CDF of the number of requests for unique domain names in the SOC dataset. A small fraction of domains is responsible for most (repeated) searches by analysts. The dashed lines separate the two clusters in Figure 1.

TABLE 1. Overview of the 78,555 requests over 15,443 unique domains in our dataset. 19.1% of domains with 5 or more search requests cause 73.5% of the total search volume.

	# Domains	%	# Requests	%
5 or more requests	2,948	19.1%	57,725	73.5%
4 or fewer requests	12,495	80.9%	20,830	26.5%
Total	15,443	100.0%	78,555	100.0%

While threat scores are provided by some threat intelligence providers [36], they do not necessarily determine the next action. For example, a domain may be associated with a worm for which a patch has already been deployed, so it poses no imminent threat. Similarly, threat intelligence databases suffer from false positives and false negatives. Hence, investigating domain names while assessing an alert requires judgement by human analysts.

III. OBSERVATIONAL STUDY

To better understand how much time analysts spend on investigating domains, and whether there is room for improvement, we received access to the logs of the domain checker tool, which analysts use to engage with threat intelligence services when investigating the domains that show up in alerts (*(5)* in Figure 1). The dataset contains requests from SOC analysts to the threat intelligence service and the timestamps of these requests in one SOC of our target MSSP. Note that this data does not contain any sensitive information about the customer (e.g., customer names, IP addresses, and device information). We have collected these logs from April 22, 2019, to October 11, 2019, for more than five consecutive months, spanning the manual investigation of 15,443 unique domains based on 78,555 search requests by analysts.

A. REPEATED INVESTIGATIONS

When we explore our dataset by looking at the number of search requests per domain, we find that a small fraction of the domain names is causing the majority of requests, with a long tail of domains that are seen only once or twice (Figure 3). In fact, we see a power-law distribution with an almost perfect example of the 80/20 rule, see Figure 1: 2,948 domains (19.1% of all domains) triggered

TABLE 2. Results of 400 sampled domain names analyses within the case study SOC. Domains posing no or a low risk make up two-thirds (63.1%) of the domains and more than three-fourth (77.1%) of requests in our sample.

SOC Status	Final Decision	Customer Risk	# Domains	%	# Total Requests	%	Avg. # Requests
Completed	Malware	High	38	9.5%	557	6.3%	14.7
	Phishing	Med	30	7.5%	425	4.8%	14.2
	PUP	Low	51	12.8%	2,412	27.2%	47.3
	FP	No	201	50.3%	4,414	49.9%	22.0
Pending	-	-	80	20.0%	1,044	11.8%	13.1
Total			400	100.0%	8,852	100.0%	22.1

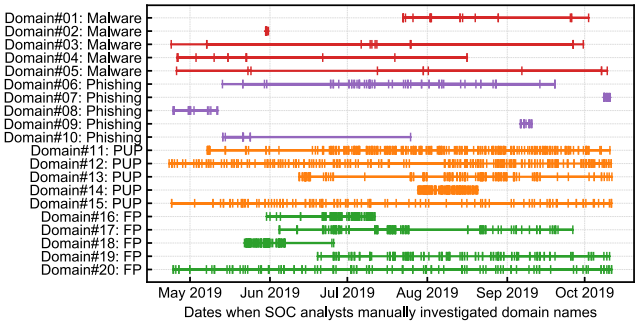

FIGURE 4. Example timelines of information requests for domains from the four threat categories in our sample. The horizontal lines indicate the time frame during which a domain occurs in the dataset. Each vertical tick marks an analyst (incompletely) investigating this domain. Note the higher density of investigations for low or no-risk domains.

57,725 requests (73.5% of all requests). In contrast, 12,495 domains were requested fewer than four times (80.9% of all domains) and they collectively triggered 20,830 requests (26.5% of all requests). This means that a majority of analysts' time is spent on looking up domains that have been looked up before without reaching a conclusion, or in parallel by multiple analysts investigating different alerts. If an analyst would complete the investigation of such a domain, the result would be recorded, and either the SIEM could directly handle alerts related to that domain, or the result from the first investigation would be available to the analyst handling an alert containing the concerned domain name. To the best of our knowledge, this distribution—and the pattern of repetitive investigations that it implies—has not been reported in earlier studies on SOC automation.

B. SAMPLE OF MOST-INVESTIGATED DOMAINS

Next, two researchers authorized to collect data inside the secure SOC environment investigated a sample of 400 domains from the total set of 2,948 domains which were investigated more than five times (19.1% of all domains in the dataset). For these domains, they assessed the final verdicts (malicious/benign/etc.) as of May 2020, i.e., more than half a year after the measurement period. We then aggregated the results on-site to contain only the domain names, their SOC status (investigation completed or pending), and their final evaluation by the SOC. The resulting dataset does not include any information identifying SOC customers or their IT environments, even for the authors involved in the study, see Figure 2.

In our sample of 400 domains, the investigation of 320 (80%) was *completed*, while the remaining 80 domain names (20%) were still *pending*. Of these 80 domains, 19 were still pending because they are non-existent or invalid domain names, for example, a string that is not a domain name entered by the user in the URL bar of the browser, or a non-existent domain name. For the remaining 61 domains, no traces were observed at all (e.g., in passive DNS, threat intelligence, and search engine), and no additional alerts had been triggered after some time. Hence we did not reach a verdict.

Next, for the 320 *completed* investigations, the SOC's final decision can be put into four major categories: malware, phishing, Potentially Unwanted Program (PUP), and False Positive (FP):

Malware includes downloads from domain names where malware is located, communication with domains used for Command and Control (C&C), and domain names used by known targeted attacks. As these indicate a compromise, we consider them *high* risk.

Phishing includes domain names of phishing sites that mimic legitimate sites to which customers are drawn via spam emails or online advertisement networks. As these domains may lead to fraud and disclosure of authentication information, but do not indicate ongoing compromises, are of *medium* risk.

Potentially Unwanted Programs differ per customer, and may include everything from small games to browser toolbars. How to handle PUPs depends on each customer's internal policies, but from a security perspective, they are a *low* risk.

False Positives occur when a security appliance flagged a domain as risky, but the investigation found it unrelated to malicious or unwanted activities. These are rated as *no risk* for the customers.

C. EVALUATION OF MOST-INVESTIGATED DOMAINS

In our sample of domains consuming most of SOC analysts' time, we find that around half (50.3%) of all domain names are false positives. A further 12.8% of domains relate to PUPs. Only 17.0% of domains in our sample relate to high or medium risks (9.5% to malware and 7.5% to phishing). Note that this is in itself not surprising. According to a recent study [14], false positive alerts in organizations account for more than a quarter of all security alerts. However, this imbalance further increases when we look at the number of requests by analysts. FPs, on average, see around 1.5 times as many requests as phishing and malware domains, while PUP-related domains may see more than three times the request volume in comparison to malware domains. In summary,

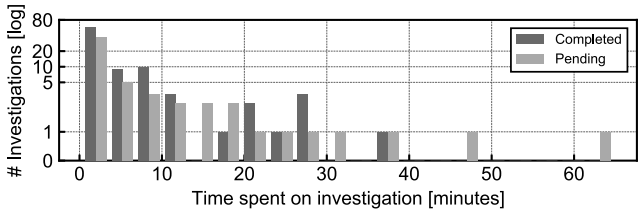

FIGURE 5. Histograms of the time taken to investigate domain names by SOC status in the case study SOC. The y-axis is on a log scale. The average time spent on completed investigations is median 3.7 (IQR: 2.7–8.3) minutes, and spent on pending domains is median 4.2 (IQR: 2.9–9.2) minutes.

77.1% of requests in our sample are caused by repeated lookups for low or no-risk domains.

We visualize example cases for the lookup patterns we see in these five categories in Figure 4. Each line is a domain and each tick signifies the time when that domain was investigated by an analyst. While the length of the investigation period varies across all categories, the higher frequency of lookups for FP and PUP related domains is clearly visible. If a verdict was reached for FP and PUP domains earlier, or if re-investigations would only be triggered when new information becomes available, significant time spent by analysts could be saved.

Hence, we suggest that the SOC should provide an indicator for domains that have a higher probability for repeated investigations in the future. This way, analysts can either attempt to finish the domain investigation (if enough information is available), or close an unsuccessful investigation noting insufficient information being available. In the former case, the SIEM can then automatically annotate and handle these domains, while in the latter case, the SIEM can take care of prompting analysts to investigate this domain *only* if new information became available in the used threat intelligence database. While this approach to give priority to domains that are likely associated with a low risk or even no risk at all might seem counter-intuitive, it *frees time* for the SOC. Our approach limits time spent on futile re-investigations of domains where the information basis did not change, and on domains that are tangential to an alert and potentially do not warrant a full investigation (which analysts only notice while already investigating it). Instead analysts can focus on *thoroughly* investigating high severity alerts and performing in-depth malware analysis.

D. TIME SPENT ON INVESTIGATIONS

In order to estimate the impact of the repeated investigations on scarce resources, we explore the amount of time SOC analysts spend on a single domain investigation. While the SOC environment operates various kinds of logging, none straightforwardly captures the duration. The start time of an investigation is observed from the query to the Domain Checker tool. The end time is more problematic, however. For completed evaluations, we can use the timestamp for when the evaluation result was entered into the alert monitor.

TABLE 3. Requests over unique domains in SOC#2 and SOC#3. Again, we find that a small set of domains with 5 or more requests is responsible for the bulk of all requests.

SOC#2 Dataset	# Domains	%	# Requests	%
5 or more requests	1,981	27.1%	90,312	92.6%
4 or fewer requests	5,329	72.9%	7,236	7.4%
Total	7,310	100.0%	97,548	100.0%

SOC#3 Dataset	# Domains	%	# Requests	%
5 or more requests	4,746	18.9%	249,577	81.1%
4 or fewer requests	20,300	81.1%	58,037	18.9%
Total	25,046	100.0%	307,614	100.0%

When investigations are incomplete, analysts often do not log any result, so we lack an event that marks the end time. Still, for a small subset of incomplete investigations, the analyst enters a 'pending' comment into the monitor. That gives us an approximate end time for those investigations.

We obtained detailed logs for 2,027 investigations by analysts at the SOC on December 6–12, 2020. For 146 investigations, we have both start and end timestamps. Of these, 88 logs have the SOC status of completed and the remaining 58 logs have the SOC status of pending. Figure 5 shows the histograms of these results by SOC status (completed or pending). Across all 146 investigations the median duration is 3.8 minutes per case, with outliers going to beyond one hour (interquartile range (IQR): 2.8–8.7 minutes). We also see that investigations resulting in 'pending' take more time than those with that are completed: a median of 4.2 (IQR: 2.9–9.2) minutes versus 3.7 minutes (IQR: 2.7–8.3) minutes, respectively. Given the volume of investigations, these estimates underline that preventing repetitive investigations is worth pursuing.

E. REASONS FOR REPEATED INVESTIGATIONS

We conducted several informal conversations with SOC analysts to understand the reasons for the repeated, incomplete investigations. Two key factors were mentioned: limited time and limited information during an investigation. How do these factors impact our idea to prioritize the investigations of domains that are likely to trigger many additional investigations?

The first factor, limited time, can be taken into account in a straightforward manner. More time could be allotted for priority investigations, since this would prevent many recurring later investigations, producing an overall efficiency gain.

The second factor, limited availability of information, poses a more difficult challenge. Missing information is not something that can be changed at the time of the investigation. That being said, almost all investigations occur under the condition of limited information. Are the domains that are repeatedly investigated suffering from lower information availability than those where the investigation is successfully

completed? Surprisingly, the answer appears to be: no. To see this, we take another look at the 146 domains we discussed in Section III-D. Of the 146 domains, 119 (82%) were investigated for the first time in the SOC. Some of these domains showed up in subsequent alerts. During that time, the analysts did not have access to the completed evaluation results because of issues with the SIEM. This meant that those domains would be investigated again, even if their prior investigation had been successfully concluded. This temporary glitch allows us to investigate if domains that are repeatedly investigated are more difficult to evaluate. Of the 119 first-time investigations, 69 (58%) were completed by analysts and 50 (42%) remained pending. We observe that in both groups, domains have the same probability of being investigated repeatedly later. Of the 69 domains 'completed' on the first attempt, 7 (10%) were investigated repeatedly, and 62 (90%) were not. Of the 50 domains 'pending' after the first attempt, 6 (12%) were investigated repeatedly, and 44 (88%) were not.

In short, whether a domain triggers subsequent investigations is independent of how difficult it is to evaluate it. Thus, it makes sense to prioritize these domains. This even holds if their investigation cannot be completed in the first attempt. If the analyst succeeds only on the second or third attempt, this will still reduce subsequent investigations.

F. GENERALIZATION TO OTHER SOCs

To determine whether the pattern of repeated investigations can be found in other SOCs as well, we also collected and examined datasets from two different SOCs (SOC#2 and SOC#3 in Figure 3). SOC#2 is also an MSSP, but operating on a different continent. It uses a different SIEM implementation and toolchain, while the basic workflow is the same as in our case-study (Figure 1). SOC#3 is an in-house SOC focused on threat hunting in the network of the organization. It is located in the same country as our case study. For SOC#2, we collected data on 97,548 investigations of 7,310 domain names by analysts between April 22, 2019 and March 26, 2020. For SOC#3, we collected data on 307,614 investigations of 25,046 domain names by analysts between June 28, 2019 and November 12, 2020. In both SOCs, we find a very similar lookup pattern for domains as we saw in our case study. Specifically, in SOC#2, 1,981 (27.1%) domain names accounted for 90,312 (92.6%) requests, and in SOC#3, 4,746 (18.9%) domain names accounted for 249,577 (81.1%) requests. These results demonstrate that the pattern of repeated investigations likely generalizes beyond our SOC to other SOC implementations, analysis tools and workflows.

IV. TOOL DESIGN

To prevent time-sinks in SOC activities, we design DomainPrio. DomainPrio predicts—in real-time—which domains will end up consuming the most SOC resources in the future. This allows analysts to prioritize reaching a conclusion on these domains, which prevents repeated future evaluations.

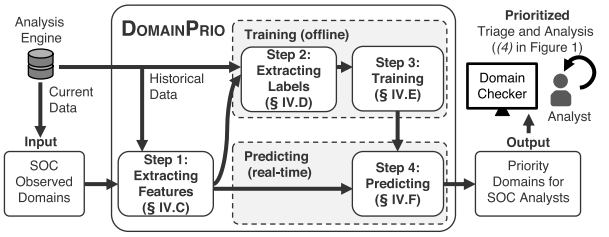

FIGURE 6. Overview of DomainPrio's steps to identify domains likely to cause repeated lookups: *(i)* extracting features, *(ii)* extracting labels, *(iii)* training, and *(iv)* predicting.

A. IMPROVEMENTS AND EFFICIENCY GAINS

The underlying assumption of the tool is that preventing continuous incomplete evaluations can save analysts time. Intuitively, and following our observations in Section III-E, there are two classes of incompletely investigated domains: i) Domains for which the investigation time is insufficient, or that are tangential to the investigation, and, ii) Domains for which not enough information is available at the time of investigation.

If we flag a domain as likely causing repeated investigations, we can instruct the analyst handling the domain to take this into account. They can then either reach a final verdict, especially on tangentially related or PUP and FP domains, or determine that there is insufficient information in threat intelligence databases to complete the investigation *at this time*. With the domain being correctly flagged, the SIEM can then automatically handle associated alerts with the Analysis Engine changing the conditions under which alerts are triggered once an evaluation is complete. For malicious domain names (related to malware or phishing), we can create custom blocklists or custom signatures, while domains identified as FPs lead to the SIEM logic to be updated to prevent alerting. If, however, the analyst annotates a domain as not having sufficient information in the available threat intelligence databases, the SIEM can automatically check whether the amount of available information, i.e., number of entries related to the domain from threat intelligence, (significantly) changed since the last investigation. If no or limited new information became available, we can alert the analyst to this, preventing them from spending time on an necessarily incomplete investigation again. In case more information became available, the analyst can reevaluate the domain. Hence, our approach can reduce the number of investigations by preventing futile investigation attempts as well as ensuring seemingly tangential time-sinks are completely investigated, preventing 100s of incomplete investigations, which we see for 19% of all domains.

Of course, one cannot evaluate a domain once and then never again, as the threat landscape changes all the time. Any SOC that records the results from previous investigations and then uses these results to handle an alert with the same threat indicator, will have to set a policy for when the initial evaluation needs to be revisited. Simply put, the SOC will

have to set a Time-To-Live (TTL) value for its completed evaluations. Our tool also functions in this context and will thus need to include the TTL policy as a setting for when a domain needs to be re-investigated. As this is part of the tool's configuration, we did not include this element in our tool.

B. APPROACH

DomainPrio consists of four steps, see Figure 6: *(i)* extracting features (Section IV-C) *(ii)* extracting labels (Section IV-D) *(iii)* training (Section IV-E) *(iv)* predicting (Section IV-F) We first automatically extract features and labels with which the model can be trained (top half of Figure 6). We then use this training data (features+labels) to generate the model. Next, the trained model is fed with the most recent extracted features of the SOC's engagements with domains to predict which observed domains would end up consuming the most SOC resources in the future (bottom half of the figure). These steps can be conducted daily to always train on the most recent data.

DomainPrio leverages existing data in the SOC environment and behavioral features of SOC analysts (e.g., how they consume threat intelligence information, when they query specific information, and how often they manually query information) to predict how time-consuming a domain name will end up being, without explicit manual labeling by an analyst. As discussed by [35], new tools added in SOC tend to be individual solutions that consume time and resources for maintenance. In contrast, we designed DomainPrio to be maintenance-free for analysts.

C. STEP 1: EXTRACTING FEATURES

To predict which domains will end up consuming the most time of SOC analysts, DomainPrio extracts features on how analysts have interacted with that domain in a configurable prior period. The data captures how the overall population of SOC analysts has interacted with the threat intelligence platform (i.e., *(5)* shown in Figure 1) after they receive domain names marked as suspicious in alerts coming from their customers' networks. Specifically, we use 80 historical SOC behavioral features summarized in Figure 4. These 80 features are calculated for 5 different time windows (i.e., 1, 3, 7, 14, and 30 days) as illustrated in Figure 7. We set up the 5 time windows because we intend to distinguish between domain names in alerts that appear in bursts over a short period of time and domain names in alerts that appear over longer periods of time. We split the most recent week into 1, 3, and 7 days, because DomainPrio is used in real time and we would like to include separate features for more granular periods. While SOCs will be different in the implementation of their systems, these features are based on data that should be readily available from logs that are present in most if not all SOCs. To the best of our knowledge, these features are novel and have not been used before to improve SOC automation.

TABLE 4. List of historical SOC behavioral features.

No.	Feature
1–5	# of requests to threat intelligence platform in the last 1/3/7/14/30 days
6–10	Elapsed days from first investigation date in the last 1/3/7/14/30 days
11–15	Elapsed days from last investigation date in the last 1/3/7/14/30 days
16–20	Investigation period in the last 1/3/7/14/30 days
21–45	Mean/Min/Max/SD/Var of # of requests per a day in the last 1/3/7/14/30 days
46–50	Weekday ratio of requests in the last 1/3/7/14/30 days
51–55	Day-shift ratio of requests in the last 1/3/7/14/30 days
56–80	Mean/Min/Max/SD/Var of interval between requests in the last 1/3/7/14/30 days

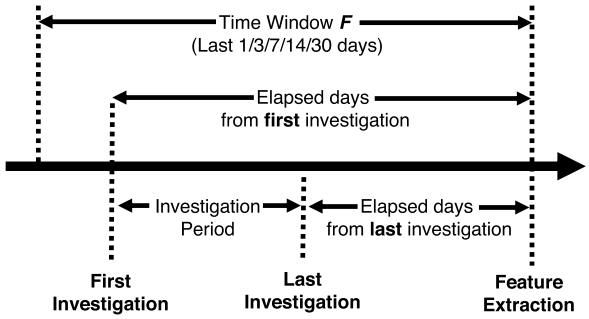

FIGURE 7. Timeline for extracting historical SOC features.

1) NUMBER OF REQUESTS TO THREAT INTELLIGENCE PLATFORM (NO. 1–5)

These features are the number of requests of a domain name by SOC analysts. Specifically, we count the number of request queries to threat intelligence platforms (e.g., VirusTotal) for each specified time window (1/3/7/14/30 days). The intuition behind these features is that suspicious domain names manually investigated by SOC analysts have different characteristics. For example, some domain names are simultaneously observed in many customers' networks in the short term and others are occasionally observed in multiple customers in the long term.

2) ELAPSED DAYS FROM FIRST INVESTIGATION DATE (NO. 6–10)

We calculate the number of days that elapse between the date of the *first* investigation by SOC analysts and that of the feature extraction in each specified time window (1/3/7/14/30 days). The intent of using these features is to distinguish between domain names that have been investigated earlier by SOC analysts and those that have been investigated more recently.

3) ELAPSED DAYS FROM LAST INVESTIGATION DATE (NO. 11–15)

Similar to the former features, we then calculate the number of days that elapsed from the date of the *last* investigation in each time window. We include this features to assess whether a domain name has been continually investigated until recently in the SOC.

4) INVESTIGATION PERIOD (NO. 16–20)

This feature is a period (days) from the first investigation to the last investigation in each time window. This is useful for distinguishing between domain names that have been investigated for a long time by SOC analysts and those that have only been investigated for a short time.

5) MEAN/MIN/MAX/SD/VAR OF NUMBER OF REQUESTS PER A DAY (NO. 21–45)

These features calculate 5 different statistics, i.e., mean, minimum, maximum, standard deviation (SD), and variance (Var), for the number of requests per day for each time window. We adopt these features to capture the trend of how multiple SOC analysts were investigating a domain within each time window.

6) WEEKDAY RATIO OF REQUESTS (NO. 46–50)

We calculate the fraction of weekdays for each date on which SOC analysts issued a request to a threat intelligence platform for each time window (the last 1/3/7/14/30 days). A weekday is defined as Monday through Friday in local time at the SOC analysts' office. The intent of designing this feature is that on weekdays SOC analysts typically investigate suspicious domain names recently observed from customer networks, however, on weekends, they tend to investigate more unusual domain names or perform threat hunting which cannot be done on weekdays due to the higher alert load then.

7) DAY-SHIFT RATIO OF REQUESTS (NO. 51–55)

Next, we calculate the fraction of requests issued by day-shift SOC analysts for each time window. There are 2 types of SOC analysts, e.g., day-shift and night-shift, to deal with customers 24 hours a day and 365 days a year. The intuition behind adopting this feature is that, as with the weekday ratio described above, the MSSP's customers tend to be large companies having most alerts during the day, thus trends differ between day-shift night-shift analysts.

8) MEAN/MIN/MAX/SD/VAR OF TIME INTERVAL BETWEEN REQUESTS (NO. 56–80)

These features calculate 5 different statistics, i.e., mean, min, max, SD, and Var, for the time interval (seconds) between requests for a domain name from SOC analysts for each time window. The reason for this is that we would like to distinguish between a request of domain names used in targeted attacks against a particular company and those used in non-targeted attacks that reach multiple companies. For example, a domain name used for a targeted attack is observed in only a few companies/employees thus rarely investigated, a domain name used for non-targeted attacks is widely spread across companies/employees thus often investigated by multiple SOC analysts in a short period.

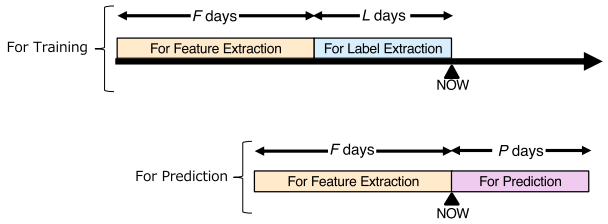

FIGURE 8. Timeline of Feature and Label Extraction. *F*-Days are used for feature extraction, *L*-Days for label extraction, *P*-Days for prediction.

D. STEP 2: EXTRACTING LABELS

The second step of DomainPrio is labeling the domain names. Our aim is not to label malicious domains, but to label domains that have consumed the most SOC resources. Our aim is that by training the model on these labels, it can predict in real time which of the more recent domains will trigger five or more future investigations. More formally, we label those domains as a *priority* that triggered more than K investigations in the L days before training the model (Figure 8). These labels can be extracted automatically from the SOC dataset. When deploying our tool in the SOC environment, we set $K = 5$. Our choice of K is based on the results of our observational study, where this count almost perfectly demarcates the 80/20 boundary. Domains with 5 or more investigations make up just 19% of the domains while triggering 74% of all investigations (Section III). By repeating label extraction on a daily basis, we can continue to add new labeled data. These labels are derived from normal operations and do not require any analyst effort.

E. STEP 3: TRAINING

We now train the model that is the core of DomainPrio. The training data consists of the extracted features and labeled domains obtained in Steps 1 and 2. We decided to employ a standard logistic regression algorithm for two reasons. First, the logistic regression is scalable and fast so that we can use it for real-time prediction of lots of suspicious domain names from a lot of customers in our SOC environment. If the algorithm is not fast enough, DomainPrio will not help SOC analysts to reduce their workload in any way. Second, the logistic regression has good interpretability, i.e., its results are explainable to SOC analysts. The output of logistic regression can be interpreted as the probability that the input domain name will be a priority and shows how each feature contributes to the result, increasing trust in the tool's recommendations.

We use an L_1 regularized logistic regression [18]. Given a domain name's feature vector $\mathbf{x}$ shown in Section IV-C, we model the conditional probability of the label $y \in \{0 \text{ (non-priority)}, 1 \text{ (priority)}\}$ with the following equation:

$$p(y = 1|\mathbf{x}; \theta) = \sigma\left(\theta^\top \mathbf{x}\right) = 1/\left(1 + \exp\left(-\theta^\top \mathbf{x}\right)\right)$$

where θ is the parameter of the logistic regression model, σ is the sigmoid function, and all features in $\mathbf{x}$ are normalized into the range [0,1]. For training, we use a set of n labeled training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ to find the parameter θ when minimizing the next objective function that introduces a hyper-parameter λ that determines the degree of regularization.

$$\min_{\theta} \sum_{i=1}^{n} -\log p\,(y_i|\mathbf{x}_i;\theta) + \lambda\|\theta\|_1$$

The L_1 regularized component ($\lambda\|\theta\|_1$) adds a penalty to the objective function. This contributes to output sparse feature weights, i.e., features that do not contribute significantly are identified and pruned. Note, that the L_1 regularization is more powerful than other regularization methods (e.g., L_2 regularization) in reducing the feature coefficients closer to zero. In other words, more features are pruned and the reduced number of features contributes to preventing overfitting. It also helps to reduce memory usage and makes the results presented to SOC analysts more concise and explainable.

F. STEP 4: PREDICTING

We use the model generated in Step 3 to predict analyst effort-intensive domain names in our SOC environment. Our goal here is to predict the probability that SOC analysts will manually analyze a domain name more than K times within P days, using only the features available at that time. Using the parameters θ trained above, the probability p that a domain name's feature vector $\mathbf{x}$ indicates effort intensity and the predicted label $\hat{y}$ are defined as follows:

$$\hat{y} = \begin{cases} 1, & \text{if } p(y|\mathbf{x};\theta) \geq 0.5 \\ 0, & \text{otherwise.} \end{cases}$$

Note that the model predicts the probability of a new domain name being repeatedly looked up using only the features available at that time.

After Step 4, DomainPrio outputs the domain names predicted to be lead to frequent domain lookups ($\hat{y} = 1$), i.e., domains that should be prioritized to prevent these repeated lookups, in descending order of their probability and presents them to SOC analysts, as shown in Figure 6. SOC analysts can then use the results to prioritize and efficiently conduct triage and deeper analysis (*(4)* in Figure 1). Specifically, SOC analysts will be required to determine and record the action to be taken on these domain names in the analysis platform (completing the investigation or marking it as 'insufficient threat intelligence'), rather than pending the decision. This ensures that domain names causing time-sinks are either handled early, avoiding *future* repeated lookups, or are only re-investigated as soon as the information basis changed, thereby increasing the SOC's efficiency.

G. DATA FLOW IN PRODUCTION

When deploying DomainPrio in production, the training phase (Steps 1–3) is conducted offline and repeated daily. It is based on the historical data of the past $F + L$ days. Each day, the newly-trained model will be deployed at the point marked "NOW" in Figure 8 and make predictions in real time. The daily update of the model is necessary in order to catch up with the changes in features due to the changes in analyst behavior after the deployment of DomainPrio. Besides, there is no downside to keeping the training model updated as it is a very lightweight process anyway. This is also supported by the low requirement of computational resources for DomainPrio in comparison to, e.g., Deep Learning based approaches, due to our choice of an L_1 regularized logistic regression.

DomainPrio receives from the SOC analysis engine a domain name in an alert from the customer environment. The domain name is processed in Step 1, and the features in Figure 4 are immediately extracted using the last F days of data from the NOW point. This is also a very lightweight process. Then, using recent features, Step 4 is performed to calculate the probability p that the domain name will be investigated more than K times in P days in the future from "NOW" on. These three steps are repeated for every incoming domain, and a set of priority domain names for SOC analysts is presented in the interface of the Domain Checker (see Figure 1).

V. TOOL EVALUATION

We evaluate whether DomainPrio can proactively identify domain names that are repeatedly investigated in a production setting. We use a dataset collected from the SOC between May 29, 2019 – October 11, 2019. Here, we follow the practices of prior studies [8], [13], [21], [34], [40] and evaluate the tool in an offline setting based on a real-world historical dataset.

A. EVALUATION SETTING

As described in Section IV-F, our goal is to predict the domain names that SOC analysts will investigate more than K times during P days. In our case study SOC, we set $K = 5$ based on the results of the observational study in Section III.

Instead of evaluating all the data across the whole dataset in one shot, we create 100 slices of sliding time windows—i.e., different overlapping time periods. This is a best practice [6], [23], [32] for evaluating machine learning systems over time. During these tests, we also evaluate the impact of the three parameters that can be supplied to our tool (the number of days for F, the feature extraction period, L, the labeling period, and P, the prediction period) on the accuracy of DomainPrio's predictions.

Figure 5 shows an example of the slices for our experiments for $F = 30$, $L = 3$, and $P = 3$. In this case, we designate the first 30 days of the dataset as training data followed by a labeling period of three days ($L = 3$). If F would be smaller (e.g., $F = 3$), then we only use the 32 features of time windows ≤ 3 days, see Section IV and Figure 4. We use the following day for real-time predicting, i.e., the day on which the model is trained and on which we assess the probability of domains in alerts being frequently

TABLE 5. Example of how we create 100 slices from our dataset for evaluation, with $F = 30$ and $L = P = 3$.

Test #	Feature Extraction Period (F = 30 days)	Labeling Period (L = 3 days)	Model Deployment Date	Real-time Predicting Period (1 day)	Prediction Period (P = 3 days)
1	2019/05/29 00:00 – 2019/06/27 23:59	– 2019/06/30 23:59	2019/07/01 00:00	2019/07/01 00:00 – 2019/07/01 23:59	– 2019/07/04 23:59
2	2019/05/30 00:00 – 2019/06/28 23:59	– 2019/07/01 23:59	2019/07/02 00:00	2019/07/02 00:00 – 2019/07/02 23:59	– 2019/07/05 23:59
3	2019/05/31 00:00 – 2019/06/29 23:59	– 2019/07/02 23:59	2019/07/03 00:00	2019/07/03 00:00 – 2019/07/03 23:59	– 2019/07/06 23:59
⋮	⋮	⋮	⋮	⋮	⋮
98	2019/09/03 00:00 – 2019/10/02 23:59	– 2019/10/05 23:59	2019/10/06 00:00	2019/10/06 00:00 – 2019/10/06 23:59	– 2019/10/09 23:59
99	2019/09/04 00:00 – 2019/10/03 23:59	– 2019/10/06 23:59	2019/10/07 00:00	2019/10/07 00:00 – 2019/10/07 23:59	– 2019/10/10 23:59
100	2019/09/05 00:00 – 2019/10/04 23:59	– 2019/10/07 23:59	2019/10/08 00:00	2019/10/08 00:00 – 2019/10/08 23:59	– 2019/10/11 23:59

looked up in the three days ($P = 3$), starting at midnight. Note that the tests are independent, i.e., no information from Test #1's model is carried over to that trained and evaluated in Test #2.

DomainPrio—including feature extraction for the logistic regression—does not require special computational machinery. Instead, it is being run on a standard laptop (1.3GHz Intel Core m7 CPU/8GB Memory/512GB SSD Disk), providing real-time performance. This is because the logistic regression algorithm used in DomainPrio and calculating features are both lightweight and scalable. This also means that our tool does not require a GPU to reach sufficient performance, as it is common with deep neural networks.

B. PERFORMANCE EVALUATION

We use the true positive rate (TPR), true negative rate (TNR), and balanced accuracy (BAC) as metrics for performance. Here, the TPR is the fraction of correctly identified priority domain names, TNR is the fraction of correctly identified non-priority domain names, and BAC is the mean of TPR and TNR, which avoids overstated performance on an imbalanced dataset to fairly evaluate the training model [16]. Correctly identified means that the model prediction for the prediction period P was correct. Note, again, that our tool does not try to identify *malicious* domain names, but rather domain names that will trigger many investigations, so that those can be prioritized and save time of subsequent investigations. Thus, positive means a domain having priority, and negative means that a domain has no priority.

First, we evaluate DomainPrio by varying the feature period F and the labeling period L, with P set to the same duration as L, see Figure 6. We find that, apart from a very short labeling and prediction period with $L = P = 1$, a larger F increases the BAC, with the highest BAC being 0.892 with $F = 30$ and $L = 7$. While an increase in the size of the feature set can be expected to improve the predictive power of our tool, we nonetheless opted to test this first, before relying on $F = 30$ in our remaining tests. We found the standard deviation (SD) to be small, especially with $L = P \geq 7$. This means that the accuracy of DomainPrio is stable over time, e.g., across changes in the threat landscape or specific analysts across the 100 slices that make up the evaluation dataset.

Next, we set $F = 30$ and evaluate DomainPrio given a varying labeling period L and prediction period P, see Figure 7. We find the BAC to peak for $L = P$, regardless of the value for L. For example, when $L = 7$, the BAC increases from $P = 1$ to $P = 7$, and then decreases from $P = 7$ to $P = 30$. Hence, we recommend $L = P$ when deploying our tool in production. However, note that as $L = P$ was increased, the best results for the TPR occur when $L = P = 30$, and for the TNR and BAC when $L = P = 7$. Given that an increase in L increases the time it takes for the most recent data to be labeled and applied to the training model (see Figure 8), and based on discussions with SOC analysts on the practicality, we hence recommend $L = P = 7$ for production use, as it has the highest BAC and a relatively small labeling period. In short, with these parameters, our model predicts with 89% accuracy which domains will consume more than 5 investigations.

C. SELECTED SOC FEATURES IN DomainPrio

Now that we know that our model achieves high accuracy, we investigate which of the SOC behavioral features shown in Figure 4 were the significant contributing features selected by DomainPrio. As explained in Section IV, DomainPrio's L_1 regularized logistic regression prunes features that do not contribute to the results and selects only those features that contribute significantly. We can inspect the coefficients of the trained logistic regression model to identify those features that increased or decreased the probability of being classified as a priority. We do this for 100 tests run with $L = P = 7$ and $F = 30$, i.e., generating 100 independent regression models. Figure 8 summarizes the most frequent features increasing/decreasing the probability across 100 test slices.

We find three notable characteristics among the features that increased the probability.

1) **Requests to threat intelligence platforms** (Features 1 and 5) were selected every time. This follows the intuition that domain names which have been investigated more often in the most recent time window have a higher probability in subsequent periods.
2) **The (SD of) requests per day** (Features 44 and 39) were also frequently selected, i.e., 100 and 92 times, respectively. This means that if there is greater variability in the number of requests per day contained in the time window (e.g., a greater difference between a well-surveyed day and a less well-surveyed day), then the domain name will have more priority later on.

TABLE 6. TPR/TNR/BAC for DomainPrio with varying F days and $L = P$ days. Mean±SD across all 100 tests.

		$L = P = 1$	$L = P = 3$	$L = P = 7$	$L = P = 14$	$L = P = 30$
$F = 1$	TPR	0.702±0.295	0.738±0.199	0.675±0.142	0.621±0.124	0.585±0.139
	TNR	0.902±0.049	0.862±0.080	0.852±0.087	0.844±0.090	0.812±0.107
	BAC	0.801±0.149	0.800±0.103	0.764±0.072	0.733±0.065	0.699±0.058
$F = 3$	TPR	0.705±0.306	0.810±0.157	0.823±0.096	0.806±0.090	0.765±0.103
	TNR	0.909±0.046	0.892±0.053	0.905±0.041	0.896±0.049	0.895±0.050
	BAC	0.806±0.153	0.851±0.077	0.864±0.047	0.851±0.046	0.830±0.050
$F = 7$	TPR	0.694±0.308	0.822±0.169	0.862±0.086	0.856±0.076	0.839±0.066
	TNR	0.912±0.043	0.915±0.040	0.913±0.038	0.905±0.040	0.892±0.052
	BAC	0.802±0.160	0.868±0.087	0.887±0.045	0.880±0.043	0.865±0.044
$F = 14$	TPR	0.675±0.333	0.815±0.189	0.861±0.089	0.860±0.075	0.861±0.061
	TNR	0.909±0.044	0.918±0.039	0.921±0.036	0.911±0.038	0.891±0.052
	BAC	0.792±0.171	0.866±0.097	0.891±0.047	0.885±0.041	0.876±0.040
$F = 30$	TPR	0.691±0.329	0.820±0.181	0.860±0.089	0.854±0.078	**0.866±0.063**
	TNR	0.904±0.044	0.918±0.043	**0.923±0.038**	0.914±0.039	0.895±0.048
	BAC	0.798±0.169	0.869±0.093	**0.892±0.048**	0.884±0.046	0.881±0.041

TABLE 7. TPR/TNR/BAC for DomainPrio with varying L and P days for $F = 30$. Mean±SD across all 100 tests.

		$P = 1$	$P = 3$	$P = 7$	$P = 14$	$P = 30$
$L = 1$	TPR	0.691±0.329	0.667±0.261	0.634±0.230	0.619±0.220	0.609±0.216
	TNR	0.904±0.044	0.900±0.047	0.871±0.051	0.861±0.052	0.862±0.051
	BAC	0.798±0.169	0.784±0.132	0.753±0.111	0.740±0.105	0.736±0.103
$L = 3$	TPR	0.661±0.162	0.820±0.181	0.777±0.117	0.739±0.121	0.719±0.117
	TNR	0.895±0.045	0.918±0.043	0.908±0.051	0.884±0.058	0.880±0.056
	BAC	0.778±0.082	0.869±0.093	0.842±0.060	0.811±0.057	0.799±0.054
$L = 7$	TPR	0.681±0.123	0.779±0.140	0.860±0.089	0.810±0.090	0.773±0.086
	TNR	0.896±0.042	0.908±0.039	0.923±0.038	**0.926±0.042**	0.904±0.047
	BAC	0.788±0.064	0.844±0.072	**0.892±0.048**	0.868±0.048	0.838±0.045
$L = 14$	TPR	0.697±0.087	0.736±0.108	0.846±0.080	0.854±0.078	0.808±0.072
	TNR	0.878±0.043	0.888±0.040	0.899±0.038	0.914±0.039	0.912±0.042
	BAC	0.787±0.049	0.812±0.060	0.873±0.046	0.884±0.046	0.860±0.041
$L = 30$	TPR	0.763±0.067	0.773±0.080	0.834±0.073	0.864±0.066	**0.866±0.063**
	TNR	0.852±0.053	0.861±0.050	0.871±0.046	0.881±0.045	0.895±0.048
	BAC	0.807±0.042	0.817±0.048	0.852±0.042	0.872±0.041	0.881±0.041

For example, a domain starting to be (incompletely) investigated more frequently on a given day will have a higher probability in the future.

3) **The time interval between requests** (Features 64, 59, 78, 69, and 57). This means that if there is greater variability in the intervals between requests in the time window, then the domain name will have a higher priority thereafter. For example, alerts that appear infrequently, but are considered to be a priority by analysts and can be investigated multiple times over a period of time, have a higher probability due to the high variability of the time interval.

Furthermore, we found three patterns among the most common features that decreased the probability:

1) **The maximum time-interval between requests** (Features 58 and 63) were both selected 98 times. This means that if the maximum time interval between requests within the time window is too large (e.g., a domain that has not been investigated in a few months), the priority of the domain name becomes lower.
2) **The time elapsed since the first/last investigation date** (Features 15 and 10) were selected 97 and 82 times, respectively. This means that if too many days have passed since a SOC analyst first/last investigated a domain name in the past, the probability of it having priority thereafter is calculated to be low.
3) **The day-shift ratio of requests** (Features 55 and 51), were selected 89 times and 79 times, respectively. This means that those domain names regularly investigated by day-shift SOC analysts are predicted to have a lower priority, and conversely, those investigated by night-shift SOC analysts will be considered to have a higher priority. For example, this indicates that domain names from alerts occurring at night, which are typically hard to observe or uncommon in customer environments, are more likely to be considered a priority and investigated by SOC analysts.

In our experiments, only 2 out of 80 features (Features 11 and 16) were not selected at least once over our 100 trial runs, with several more being rarely selected, e.g., Feature 6. On closer investigation, we found that they are features of elapsed days or periods of time calculated only over a 1-day time window. Hence, one might consider removing these features in the future.

D. REQUIRED NUMBER OF INVESTIGATIONS

The faster DomainPrio can identify priority domains, the more repeated investigations can be prevented. While DomainPrio uses a range of features (see Section IV-C), its score also depends on the number of prior investigations in order to label priority domains. From our evaluation results for DomainPrio (deployed with $L = P = 7$ and $F = 30$), we can quantify how many previous analyst investigations are required for DomainPrio to identify a domain as a priority. On production data, DomainPrio was able to identify priority domains with as little as two prior analyst engagements. While individual outliers may only be flagged as a priority after up to 82 prior analyst engagements, the median on our dataset is 6 with an IQR of 4–14.

Based on these numbers, we can then also calculate how many subsequent investigations can be prevented. This ranges from 10 to a maximum of 341 prevented investigations (median: 33.5; IQR: 17–59.8). In the best case, DomainPrio identified a domain as a priority after three investigations of the domain, which could have prevented up to 341 subsequent investigations. These numbers are several times higher than the number of required investigations, indicating the potential for large efficiency gains.

TABLE 8. Top 10 features that increase or decrease the probability of being classified as a priority.

Top N	No.	Features that increased the probability	#Times	No.	Features that decreased the probability	#Times
1	1	Number of requests to threat intelligence platform in the last 1 day	100	58	Max of time interval between requests in the last 1 day	98
2	5	Number of requests to threat intelligence platform in the last 30 days	100	63	Max of time interval between requests in the last 3 days	98
3	44	SD of number of requests per a day in the last 30 days	100	45	Var of number of requests per a day in the last 30 days	98
4	7	Elapsed days from first investigation date in the last 3 days	98	15	Elapsed days from last investigation date in the last 30 days	97
5	64	SD of time interval between requests in the last 3 days	97	55	Day-shift ratio of requests in the last 30 days	89
6	39	SD of number of requests per a day in the last 14 days	92	41	Mean of number of requests per a day in the last 30 days	85
7	59	SD of time interval between requests in the last 1 day	90	10	Elapsed days from first investigation date in the last 30 days	82
8	78	Max of time interval between requests in the last 30 days	90	51	Day-shift ratio of requests in the last 1 day	79
9	69	SD of time interval between requests in the last 7 days	88	65	Var of time interval between requests in the last 3 days	78
10	57	Min of time interval between requests in the last 1 day	88	38	Max of number of requests per a day in the last 14 days	78

E. COMPARISON: PRIORITY BY NO. OF INVESTIGATIONS

While our evaluation demonstrates that DomainPrio can effectively identify priority domains to improve the efficiency of the SOC, one might intuitively claim that the same result could be obtained when prioritizing domains purely based on the number of incomplete investigations in the past. Hence, we also compare DomainPrio's performance against that simple baseline solution: prioritizing a domain name if it has been investigated more than R times in the past. After evaluating this naive baseline tool with the same settings and dataset as DomainPrio, we found that the BAC of that baseline tool was always worse than that of DomainPrio under any R or P. Figure 9 shows the evaluation results of the baseline tool prioritizing by the number of prior requests. In fact, the best BAC for that baseline tool is only 0.799 ± 0.049 ($R = 5$, $P = 30$), while the best BAC for DomainPrio is 0.892 ± 0.048. This reinforces our findings in Section V-C. The inclusion of additional features, such as those focusing on the behavior of SOC analysts, improves the performance of DomainPrio.

F. COMPARISON: PRIORITY BY THREAT LEVEL

Finally, we test how DomainPrio fares against metrics prioritizing domains based on their estimated maliciousness, as common in the literature [13], [19], [21], [22], [26]. As we can run this analysis on our dataset in *hindsight*, we do not have to rely on in-place calculated metrics for maliciousness. Instead, we can use VirusTotal [36]—an established tool in the field [24] integrating data from over 60 security vendors [37]—as the primary threat intelligence source, to assess which threat level domains ideally would have received when they were first encountered.

Our baseline maliciousness score tool first checks VirusTotal's domain report [38] to see whether the domain name is associated with known threats. It then counts the number of times a domain name is considered malicious by engines in the VirusTotal dataset across four threat types:

1) **Detected URLs:** The number of malicious URLs that contain the domain name
2) **Detected communicating files:** The number of binaries seen connecting to the domain
3) **Detected downloaded files:** The number of malicious files retrieved from this URL
4) **Detected referring files:** The number of malicious files that contain the domain name as a string

If the sum of these four metrics exceeds a predefined threshold (V), the baseline tool predicts the domain has priority, with a higher V indicating a higher priority. Hence, as we evaluate DomainPrio on historic data, we can benchmark it against a baseline tool that is representative of how a threat-level assessment tool would prioritize domains.

When evaluating our dataset with the baseline tool for $V = 1..V = 10$, the BAC of the baseline tool is on average 0.2–0.3 lower than for DomainPrio under similar conditions (in terms of P), see Figure 6. Figure 10 shows the evaluation results of the baseline tool prioritizing by threat level. Even in the best case for the baseline tool ($V = 4$, $P = 30$), the BAC remains at 0.609 ± 0.052, while DomainPrio reaches a BAC of 0.892 ± 0.048 in its best case ($F = 30$ and $P = L = 7$). This means that the baseline tool is not significantly better than random chance—and significantly worse than DomainPrio—at predicting whether an incoming domain name will constitute a time sink, and should therefore be analyzed for automated handling by the SIEM as early as possible. This is consistent with our finding in Section III-B that repeat investigations are worse for domains that pose no or only a low threat. To put it differently, DomainPrio is a solution that works orthogonally to threat-based automation for SOCs. One can best view them as complementary.

G. EFFICIENCY GAINS BY DomainPrio

We calculate the potential efficiency gains DomainPrio could have provided to the SOC (assuming DomainPrio was deployed with $L = P = 7$ and $F = 30$). To estimate the upper-bound efficiency gain, we assume that if a domain was identified as a priority, its investigation would be completed on the day it was first observed, preventing subsequent (incomplete) investigations of that domain—the ideal scenario envisioned by DomainPrio. While this upper bound will not be reached in practice (see Section IV-A), it provides an estimate of the expected gains' order of magnitude.

From our historical dataset, we take a period of 100 days. This contains 9,668 unique domain names in 30,269 investigations by SOC analysts. In the ideal scenario, DomainPrio would have reduced the number of investigations to 19,554, i.e., a dramatic efficiency gain of 35%—calculated

TABLE 9. TPR/TNR/BAC for the baseline tool prioritizing by the number of requests with varying *R* and *P* days. Mean±SD across all 100 tests.

		$P = 1$	$P = 3$	$P = 7$	$P = 14$	$P = 30$
$R = 5$	TPR	0.800±0.160	**0.838±0.092**	**0.838±0.077**	0.818±0.066	0.790±0.059
	TNR	0.559±0.060	0.638±0.065	0.713±0.059	0.768±0.058	0.808±0.065
	BAC	0.680±0.083	0.738±0.059	0.775±0.051	0.793±0.047	**0.799±0.049**
$R = 10$	TPR	0.718±0.181	0.746±0.117	0.724±0.103	0.683±0.089	0.637±0.077
	TNR	0.690±0.063	0.774±0.057	0.842±0.046	0.884±0.044	0.905±0.047
	BAC	0.704±0.088	0.760±0.064	0.783±0.057	0.783±0.050	0.771±0.047
$R = 15$	TPR	0.650±0.204	0.666±0.130	0.628±0.117	0.572±0.102	0.521±0.086
	TNR	0.763±0.058	0.844±0.048	0.903±0.037	0.928±0.033	0.935±0.034
	BAC	0.707±0.094	0.755±0.067	0.766±0.063	0.750±0.055	0.728±0.047
$R = 20$	TPR	0.590±0.202	0.597±0.129	0.553±0.119	0.492±0.103	0.442±0.089
	TNR	0.810±0.055	0.884±0.042	0.936±0.028	0.952±0.027	0.954±0.027
	BAC	0.700±0.095	0.741±0.065	0.744±0.061	0.722±0.054	0.698±0.048
$R = 25$	TPR	0.540±0.202	0.546±0.124	0.494±0.115	0.432±0.098	0.387±0.086
	TNR	0.842±0.046	0.911±0.034	0.954±0.023	0.965±0.023	**0.966±0.023**
	BAC	0.691±0.095	0.729±0.061	0.724±0.058	0.698±0.051	0.677±0.045

TABLE 10. TPR/TNR/BAC for the baseline tool prioritizing by threat level with varying *V* and *P* days. Mean±SD across all 100 tests.

		$P = 1$	$P = 3$	$P = 7$	$P = 14$	$P = 30$
$V = 1$	TPR	0.714±0.072	0.694±0.104	0.726±0.103	0.734±0.098	**0.738±0.087**
	TNR	0.430±0.065	0.434±0.065	0.439±0.068	0.443±0.071	0.454±0.071
	BAC	0.572±0.049	0.564±0.058	0.582±0.059	0.588±0.059	0.596±0.054
$V = 2$	TPR	0.656±0.075	0.633±0.104	0.664±0.103	0.670±0.095	0.676±0.086
	TNR	0.516±0.065	0.519±0.065	0.523±0.066	0.527±0.071	0.538±0.071
	BAC	0.586±0.049	0.576±0.057	0.593±0.058	0.598±0.056	0.607±0.053
$V = 3$	TPR	0.612±0.081	0.585±0.109	0.611±0.103	0.619±0.100	0.627±0.091
	TNR	0.560±0.067	0.564±0.068	0.568±0.068	0.572±0.073	0.584±0.073
	BAC	0.586±0.050	0.574±0.057	0.589±0.057	0.595±0.058	0.606±0.056
$V = 4$	TPR	0.588±0.080	0.555±0.105	0.583±0.101	0.594±0.096	0.604±0.087
	TNR	0.591±0.067	0.595±0.067	0.599±0.068	0.602±0.073	0.614±0.075
	BAC	0.589±0.050	0.575±0.052	0.591±0.053	0.598±0.054	**0.609±0.052**
$V = 5$	TPR	0.566±0.074	0.541±0.106	0.555±0.097	0.562±0.092	0.574±0.082
	TNR	0.613±0.066	0.617±0.066	0.621±0.066	0.624±0.070	**0.636±0.072**
	BAC	0.590±0.048	0.579±0.052	0.588±0.051	0.593±0.051	0.605±0.049

as (30,269–19,554) / 30,269 * 100. How many hours this saves is, of course, dependent on the size of the SOC—more precisely, on the overall volume of investigations—where DomainPrio would be deployed. In our case study, this would translate into an estimate of time saved of median 679 hours (IQR: 500–1,554 hours), based on the median of 3.8 minutes (IQR: 2.8–8.7 minutes) per investigation (see Section III-D). Taking the IQR, for the 100 day period covered by our estimate, these savings add up to 0.6–1.9 full time positions (based on an 8-hour workday) that could be re-allocated to more valuable tasks like threat hunting.

To repeat, the SOC won't reach this upper-bound efficiency gain. However, as we discussed in Section IV-A, the limitations are unlikely to erode most of the efficiency gains enabled by the tool. It is also worth highlighting that the tool is practical, cheap and easily deployable, so these gains can be captured against a modest investment. All in all, we find that DomainPrio has the potential for dramatic efficiency gains, freeing up significant time for analysts to shift towards threat.

VI. DISCUSSION

In this section, we first discuss the implications our evaluation has for deploying DomainPrio in a SOC. Furthermore, we describe the limitations of our research and DomainPrio which have to be considered when using it in practice.

As explained in Section IV, we designed DomainPrio for actual deployment. Under these conditions, a core requirement of DomainPrio is that it should not increase the workload of SOC analysts. This means that we do not use any manually labeled data, but designed it to work out-of-the-box with already available data from the SOC. Hence, DomainPrio uses features focused on the behavior of SOC analysts when using threat intelligence services. Automatic extraction of labels using time-series data from the SOC allows the autonomous generation and deployment of a continually updated model.

A SOC implementing DomainPrio has to be aware that it is complementary to other proposed forms of automation, i.e., predicting maliciousness or enriching alerts. The output of DomainPrio is whether the domain should be a priority for analysts to complete an evaluation. It does not conflict with other tools that assist in analyzing alerts, and can be combined with them.

Unlike earlier tools that specifically work on handling or rating alerts, DomainPrio ranks existing workload items. This means that potential false negatives or false positives do not have a significant impact on the SOC performance. A false negative of DomainPrio does not lead to a threat being overlooked, but instead only to a domain potentially being investigated multiple times. The only downside is that the existing time loss associated with that process is not remediated. Similarly, a false positive will only result in analysts working earlier and perhaps a bit longer on an alert within the current SOC workflow. The main effect of false negatives and false positives is that efficiency gains of the tool will be marginally lower than the theoretical maximum, *not* that the SOC would miss major alerts or misdiagnose them. In other words, the tool does not need to meet the high level of accuracy usually required for usable machine learning (ML)-based SOC support tools.

DomainPrio is not a complex black box AI, but a straightforward logistic regression model that is likely better aligned with analyst mental models. Since our goal is to improve SOC operational efficiency, we would not want to spend a lot of computational cost and time on creating a training model of the tool itself and searching for its optimal parameters or performing yet another computation to obtain an explanation of its output results.

Our observational study suggests that, prior to deciding on the deployment of DomainPrio, SOCs could test the value of the tool by computing simple metrics for SOC management, e.g., how often an alert investigation was left

unresolved or how often the same domain or IP address was repeatedly investigated. This will help to find avenues potentially very effective forms of automation. The work of Sundaramurthy *et al.* [30] explains why these paths are often not explored and contribute to notification fatigue.

VII. LIMITATIONS

We discuss four key limitations of our research and DomainPrio. First, our evaluation of DomainPrio is based on a single SOC. Even though we find similar investigation patterns in different SOCs, both in geography, customer base and organizational setup, the model parameters that were found to be optimal in our case study might need to be adjusted for the deployment in another SOC.

Second, the threat level of a domain can change over time. Hence, also domains initially flagged as benign should later be reinvestigated. While this issue is not specific to DomainPrio, we suggested attaching a time-to-live (TTL) to domain names, after which the domain is reset to 'not investigated before' to address this issue.

Third, our tool has been tested in the context of domain names and not for other threat indicators. Hence, before deploying the same prioritization on other aspects of alerts, e.g., IP addresses or file hashes, the efficacy of DomainPrio will have to be reevaluated.

Finally, the efficiency gains of DomainPrio have been estimated based on historic data from a real SOC, and not measured after a full-scale deployment of DomainPrio in that SOC. Hence, our estimate of the efficiency gains is a theoretical maximum, see Section IV-A for a broader discussion.

VIII. RELATED WORK

Here, we briefly discuss recent examples of related work in terms of support tools that aid analysts in their SOC work and approaches to identify malicious domains.

A. SUPPORTING SOC ANALYSTS

Previous studies have focused on helping reduce the workload of SOC analysts and enhance detection capabilities for enterprise customers. These studies can be further divided into two main categories: estimating maliciousness scores for alerts and extracting relevant information from alerts. Due to the extent of literature, we focus on the most recent work here.

1) ESTIMATING MALICIOUSNESS

Hassan *et al.* [13] propose NoDoze to rank threat alerts based on their aggregate anomaly scores, reduce false alarms, and provide contextual explanations of the generated alerts. However, this and similar systems need to use OS-level logs of hosts in each enterprise and do not consider outsourced SOC environments which have many enterprises (customers). Akinrolabu *et al.* [2] propose features to identify malicious network traffic that is useful in SOC environments through interviews with SOC analysts. However, these features are focused on the behavior of malware and assess the maliciousness of alerts. Veeramachaneni *et al.* [34] propose an approach that combines security analysts' feedback/labeling and machine learning techniques to detect unseen attacks. Gupta *et al.* [11] propose an approach that uses a deep neural network to predict whether an event is identified as malicious by SOC analysts and notified to customers. However, such approaches need explicit labeling by analysts which requires additional time/budget and identifies events similar to prior malicious events. Shibahara *et al.* [28] propose a system to prioritize alerts and identify potentially successful attacks in an outsourced SOC environment by analyzing the correlation between alerts from different vendors' appliances to assess an alert threat level.

2) EXTRACTING INFORMATION FROM ALERTS

Zhong *et al.* [40] propose a system to extract rules for filter operations, search operations, and selection operations from multiple events and logs from the history of SOC analysts' SIEM operations. The same authors [39] propose a system of recording SIEM operations during senior analysts' work to bridge the skills gap between senior and junior analysts at SOCs. Chen *et al.* [8] developed a ''virtual'' security product which predicts what security events would have been generated by a security product if it had been present. The virtual product can help SOC analysts to enrich events and to make better decisions in terms of handling alerts/incidents.

3) SUMMARY

While the SOC environments and motivation of prioritization in the related work is similar to ours, the approach is fundamentally different. Related work attempts to predict the maliciousness or threat level of alerts or observed malware, similar to our baseline tool, where we do not predict maliciousness but can even use *a-posteriori* ground-truth data in our evaluation, providing reliable threat level estimates. In contrast to that we utilize information on SOC analysts' behavior to reduce their workload by detecting and preventing time sinks analysts encounter.

B. DETECTING MALICIOUS DOMAIN NAMES

Several studies focused on the difference between malicious and non-malicious domain names and their corresponding feature vectors to detect malicious ones using machine learning algorithms.

Ma *et al.* [20] detect malicious domain names and URLs based on their lexical structure. Félegyházi *et al.* [10] focus on using WHOIS information to detect malicious domain names. Notos [3] was the first domain-reputation system to detect malicious domain names that share similar patterns in terms of IP address and domain-name usage. Sato *et al.* [27] rely on DNS queries from multiple infected devices to find malicious domain names. Exposure [7] is a proposal to find malicious domain names based on the time-series changes in DNS traffic or queries. Antonakakis *et al.* [4] propose Kopis, which uses the characteristics of user behavior observed in authoritative name servers to detect malicious domain names. They also propose Pleiades [5], which focuses on

DNS queries to non-existent domain names in caching name servers to detect malicious C&C domain names. Segugio [25] was proposed to detect C&C domain names from DNS traffic patterns in large ISP networks. Chiba *et al.* [9] used time-series features of domain-name usage and network-based features of IP addresses and domain names to detect malicious domain names. Predator [12] is a system by Hao *et al.* to detect malicious domain names when they are registered. Oprea *et al.* [22] propose a system called MADE to prioritize the riskiest domain names contacted by hosts using a supervised machine learning algorithm and data extracted from security logs.

1) SUMMARY

The major objective of related work is detecting *malicious* domains in various networks to directly flag them or prioritize their investigation. Again, this is fundamentally different from our approach of prioritizing domain investigations based on what saves the SOC the most time. Hence, what ultimately makes our work unique is not using the 'maliciousness' of a domain as a proxy to estimate how worthwhile an investigation is, but directly estimating how likely it is that the SOC saves time by investigating a domain name *now* rather than (multiple times) *later*.

IX. CONCLUSION

In this work, we present a novel perspective on improving SOC efficiency. Based on an observational study of the SOC's activity logs, we find that a major time drain in SOCs is the repeated and incomplete investigation of domain names encountered during alert investigations.

To reduce this overhead, we propose DomainPrio, a tool using an auditable logistic regression model, which identifies domains that are likely to become time drains for the SOC. We use a feature set from the SOC analysts' prior engagements with similar domain names, without requiring manual labeling of input data. This enables analysts to conclusively investigate these domains as soon as they show up, which in turn enables the SIEM to directly present this information to analysts or to automatically handle the alert without analyst intervention when the domain is encountered again. Using SOC data from a production setting, we estimate that the deployment of DomainPrio can reduce the number of requests SOC analysts have to issue by up to 35% by preventing repeated and incomplete threat intelligence requests for domain names encountered in alerts.

We hope that our findings will benefit the SOC community and enable analysts to save time where possible, so they can focus on the creative and challenging analysis tasks, instead of suffering from alert fatigue.

REFERENCES

[1] E. Agyepong, Y. Cherdantseva, P. Reinecke, and P. Burnap, "Challenges and performance metrics for security operations center analysts: A systematic review," *J. Cyber Secur. Technol.*, vol. 4, no. 3, pp. 125–152, Jul. 2020, doi: 10.1080/23742917.2019.1698178.

[2] O. Akinrolabu, I. Agrafiotis, and A. Erola, "The challenge of detecting sophisticated attacks: Insights from SOC Analysts," in *Proc. 13th Int. Conf. Availability, Rel. Secur.*, C. Doerr and S. Schrittwieser, Eds., Hamburg, Germany, 2018, pp. 1–9, doi: 10.1145/3230833.3233280.

[3] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for DNS," in *Proc. 19th USENIX Secur.*, I. Goldberg, Ed., 2010, pp. 273–290. [Online]. Available: http://www.usenix.org/events/sec10/tech/full_papers/Antonakakis.pdf

[4] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, and D. Dagon, "Detecting malware domains at the upper DNS hierarchy," in *Proc. 20th USENIX Secur.*, D. Wagner, Ed., pp. 1–16. [Online]. Available: http://static.usenix.org/events/sec11/tech/full_papers/Antonakakis.pdf

[5] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in *Proc. 21st USENIX Secur.*, T. Kohno, Ed., 2012, pp. 491–506. [Online]. Available: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/antonakakis

[6] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Inf. Sci.*, vol. 191, pp. 192–213, May 2012.

[7] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding malicious domains using passive DNS analysis," in *Proc. 18th NDSS*, 2011, pp. 1–17. [Online]. Available: https://www.ndss-symposium.org/ndss2011/exposure-finding-malicious-domains-using-passive-dns-analysis

[8] S.-T. Chen, Y. Han, D. H. Chau, C. Gates, M. Hart, and K. A. Roundy, "Predicting cyber threats with virtual security products," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, Dec. 2017, pp. 189–199, doi: 10.1145/3134600.3134617.

[9] D. Chiba, T. Yagi, M. Akiyama, T. Shibahara, T. Yada, T. Mori, and S. Goto, "DomainProfiler: Discovering domain names abused in future," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, D. Cotroneo and C. Nita-Rotaru, Eds., Jun. 2016, pp. 491–502, doi: 10.1109/DSN.2016.51.

[10] M. Félegyházi, C. Kreibich, and V. Paxson, "On the potential of proactive domain blacklisting," in *Proc. 3rd USENIX LEET Workshop*, M. Bailey, Ed. Apr. 2010. [Online]. Available: https://www.usenix.org/conference/leet-10/potential-proactive-domain-blacklisting

[11] N. Gupta, I. Traore, and P. M. F. de Quinan, "Automated event prioritization for security operation center using deep learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 5864–5872, doi: 10.1109/BigData47090.2019.9006073.

[12] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster, "PREDATOR: Proactive recognition and elimination of domain abuse at time-of-registration," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, C. Kruegel and A. C. M. S. Halevi, Eds., Oct. 2016, pp. 1568–1579, doi: 10.1145/2976749.2978317.

[13] W. U. Hassan, S. Guo, D. Li, Z. Chen, K. Jee, Z. Li, and A. Bates, "NoDoze: Combatting threat alert fatigue with automated provenance triage," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, A. Oprea and D. Xu, Eds., 2019, pp. 1–15, doi: 10.14722/ndss.2019.23349.

[14] M. Hill. *Over a Quarter of Security Alerts Are False Positives*. Infosecurity Magazine. Accessed: Mar. 17, 2020. [Online]. Available: https://www.infosecurity-magazine.com/news/security-alerts-false-positives/

[15] (Jun. 2019). *Improving the Effectiveness of the Security Operations Center*. [Online]. Available: https://www.devo.com/wpcontent/uploads/2019/07/2019-Devo-Ponemon-Study-Final.pdf

[16] J. D. Kelleher, B. Mac Namee, and A. D'arcy, *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. Cambridge, MA, USA: MIT Press, 2015.

[17] F. B. Kokulu, A. Soneji, T. Bao, Y. Shoshitaishvili, Z. Zhao, A. Doupé, and A. hn.-J. andG, "Matched and mismatched socs: A qualitative study on security operations center issues," in *Proc. 26th ACM CCS*, X. Wang and J. Katz, Eds., 2019, pp. 1955–1970, doi: 10.1145/3319535.3354239.

[18] S.-I. Lee, H. Lee, P. Abbeel, and A. Y. Ng, "Efficient L_1 regularized logistic regression," in *Proc. 21st AAAI*, Y. Gil and R. J. Mooney, Eds., 2006, pp. 401–408.

[19] Y. Liu, M. Zhang, D. Li, K. Jee, Z. Li, Z. Wu, J. Rhee, and P. Mittal, "Towards a timely causality analysis for enterprise security," in *Proc. 25th NDSS*, P. Traynor and A. Oprea, Eds., 2018, pp. 1–15, doi: 10.14722/ndss.2018.23254.

[20] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious urls," in *Proc. 15th ACM KDD*, J. F. E. IV, F. Fogelman-Soulié, P. A. Flach, and M. J. Zaki, Eds., 2009, pp. 1245–1254, doi: 10.1145/1557019.1557153.

[21] P. Najafi, A. Mühle, W. Pünter, F. Cheng, and C. Meinel, "MalRank: A measure of maliciousness in SIEM-based knowledge graphs," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, G. Gu and D. D. Yao, Eds., Dec. 2019, pp. 417–429, doi: 10.1145/3359789.3359791.

[22] A. Oprea, Z. Li, R. Norris, and K. Bowers, "Made: Security analytics for enterprise threat detection," in *Proc. 34th ACSAC*, J. Caballero and G. Gu, Eds., 2018, pp. 124–136, doi: 10.1145/3274694.3274710.

[23] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro, "TESSERACT: Eliminating experimental bias in malware classification across space and time," in *Proc. 28th USENIX Secur.*, N. Heninger and P. Traynor, Eds., 2019, pp. 729–746. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/pendlebury

[24] P. Peng, L. Yang, L. Song, and G. Wang, "Opening the blackbox of VirusTotal: Analyzing online phishing scan engines," in *Proc. Internet Meas. Conf.*, P. Gill and R. Beverly, Eds., Oct. 2019, pp. 478–485, doi: 10.1145/3355369.3355585.

[25] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: Efficient behavior-based tracking of malware-control domains in large ISP networks," in *Proc. 45th IEEE/IFIP DSN*, J. Karlsson and Y. Amir, Eds., Jun. 2015, pp. 403–414, doi: 10.1109/DSN.2015.35.

[26] K. A. Roundy, A. Tamersoy, M. Spertus, M. Hart, D. Kats, M. Dell'Amico, and R. Scott, "Smoke detector: Cross-product intrusion detection with weak indicators," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, D. Balzarotti and J. Caballero, Eds., Dec. 2017, pp. 200–211, doi: 10.1145/3134600.3134645.

[27] K. Sato, K. Ishibashi, T. Toyono, and N. Miyake, "Extending black domain name list by using co-occurrence relation between DNS queries," in *Proc. 3rd USENIX LEET Workshop*, M. Bailey, Ed. Apr. 2010. [Online]. Available: https://www.usenix.org/conference/leet-10/extending-black-domain-name-list-using-co-occurrence-relation-between-dns-queries

[28] T. Shibahara, H. Kodera, D. Chiba, M. Akiyama, K. Hato, O. Söderström, D. Dalek, and M. Murata, "Cross-vendor knowledge transfer for managed security services with triplet network," in *Proc. 12th ACM Workshop Artif. Intell. Secur. (AISec)*, S. Afroz, B. Biggio, N. Carlini, Y. Elovici, and A. Shabtai, Eds., 2019, pp. 59–69, doi: 10.1145/3338501.3357367.

[29] J. Staddon and N. Easterday, "It's a generally exhausting field' a large-scale study of security incident management workflows and pain points," in *Proc. 17th Int. Conf. Privacy, Secur. Trust (PST)*, Fredericton, NB, Canada, Aug. 2019, pp. 1–12, doi: 10.1109/PST47121.2019.8949012.

[30] S. C. Sundaramurthy, A. G. Bardas, J. Case, X. Ou, M. Wesch, J. McHugh, and R. andS Rajagopalan, "A human capital model for mitigating security analyst burnout," in *Proc. 11th SOUPS*, L. F. Cranor, R. Biddle, and S. Consolvo, Eds., 2015, pp. 347–359. [Online]. Available: https://www.usenix.org/conference/soups2015/proceedings/presentation/sundaramurthy

[31] S. C. Sundaramurthy, J. McHugh, X. Ou, M. Wesch, A. G. Bardas, and S. R. Rajagopalan, "Turning contradictions into innovations or: How we learned to stop whining and improve security operations," in *Proc. 12th SOUPS*, S. Consolvo and M. Smith, Eds., 2016, pp. 237–251. [Online]. Available: https://www.usenix.org/conference/soups2016/technical-sessions/presentation/sundaramurthy

[32] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: An analysis and review," *Int. J. Forecasting*, vol. 16, no. 4, pp. 437–450, Oct. 2000, doi: 10.1016/S0169-2070(00)00065-0.

[33] (Jan. 2020). *The Economics of Security Operations Centers: What is the True Cost for Effective Results?*. [Online]. Available: https://www.ponemon.org/blog/the-economics-of-securityoperations-centers-what-is-the-true-cost-for-effective-results

[34] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li, "AI^2: Training a big data machine to defend," in *Proc. IEEE 2nd Int. Conf. Big Data Secur. Cloud (BigDataSecurity), IEEE Int. Conf. High Perform. Smart Comput. (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS)*, T. Zhang, J. Dongarra, and W. Qiang, Eds. New York, NY, USA, Apr. 2016, pp. 49–54, doi: 10.1109/BigDataSecurity-HPSC-IDS.2016.79.

[35] M. Vielberth, F. Bohm, I. Fichtinger, and G. Pernul, "Security operations center: A systematic study and open challenges," *IEEE Access*, vol. 8, pp. 227756–227779, 2020, doi: 10.1109/ACCESS.2020.3045514.

[36] *VirusTotal*. Accessed: Oct. 1, 2021. [Online]. Available: https://www.virustotal.com/

[37] *VirusTotal Contributors*. Accessed: Oct. 1, 2021. [Online]. Available: https://support.virustotal.com/hc/en-us/articles/115002146809-Contributors

[38] *VirusTotal Reports*. Accessed: Oct. 1, 2021. [Online]. Available: https://support.virustotal.com/hc/en-us/articles/115002719069-Reports

[39] C. Zhong, T. Lin, P. Liu, J. Yen, and K. Chen, "A cyber security data triage operation retrieval system," *Comput. Secur.*, vol. 76, pp. 12–31, Jul. 2018, doi: 10.1016/j.cose.2018.02.011.

[40] C. Zhong, J. Yen, P. Liu, and R. F. Erbacher, "Automate cybersecurity data triage by leveraging human Analysts' cognitive process," in *Proc. IEEE IEEE 2nd Int. Conf. Big Data Secur. Cloud (BigDataSecurity) Int. Conf. High Perform. Smart Comput. (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS)*, W. Li and Z. Fei, Eds. New York, NY, USA, Apr. 2016, pp. 357–363, doi: 10.1109/BigDataSecurity-HPSC-IDS.2016.41.

DAIKI CHIBA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in computer science from Waseda University, in 2011, 2013, and 2017, respectively. Since 2013, he has been with Nippon Telegraph and Telephone Corporation (NTT), where he has been engaged in research on cyber security through data analysis. He is currently a Senior Engineer at NTT Security (Japan) KK, Tokyo, Japan. He is a member of the IEICE.

MITSUAKI AKIYAMA (Member, IEEE) received the M.E. and Ph.D. degrees in information science from the Nara Institute of Science and Technology, in 2007 and 2013, respectively. Since 2007, he has been with Nippon Telegraph and Telephone Corporation (NTT), where he has been engaged in research and development on cybersecurity. He is currently a Senior Distinguished Researcher at NTT Social Informatics Laboratories. His research interests include cybersecurity measurement, offensive security, and usable security and privacy. He is a Senior Member of IPSJ and a member of IEICE. He received the Cybersecurity Encouragement Award of the Minister for Internal Affairs and Communications in 2020, the ISOC NDSS 2020 Distinguished Paper Award in 2020, and the IPSJ/IEEE Computer Society Young Computer Researcher Award in 2022.

YUTO OTSUKI received the B.E. degree from the College of Information Science and Engineering, Ritsumeikan University, in 2011, the M.E. degree from the Graduate School of Science and Engineering, Ritsumeikan University, in 2013, and the D.Eng. degree from the Graduate School of Information Science and Engineering, Ritsumeikan University, in 2016. From 2016 to 2019, he was engaged in research of malware analysis and digital forensics at NTT Secure Platform Laboratories. He is currently with NTT Security (Japan) KK.

HIROKI HADA received the M.E. and Ph.D. degrees in information security from the Institute of Information Security, Japan, in 2013 and 2019, respectively. He is currently a Security Engineer at NTT Security (Japan) KK, Tokyo, Japan. His research interests include reverse engineering, malware analysis, and incident response. He is a member of the IPSJ.

TAKESHI YAGI received the B.E. degree in electrical and electronic engineering and the M.E. degree in science and technology from Chiba University, Japan, in 2000 and 2002, respectively, and the Ph.D. degree in information science and technology from Osaka University, Osaka, Japan, in 2013. He joined Nippon Telegraph and Telephone Corporation (NTT), in 2002. He is currently the Director of the Research and Development Planning Department, NTT. His research interests include cybersecurity monitoring and security intelligence. He is a member of the IEEJ, IPSJ, and IEICE.

TOBIAS FIEBIG is currently a Researcher focusing on identifying and mitigating human-factors-based and preventable security issues in IT systems like those all too common in the Internet of Things. For this, he combines qualitative research methods with tools for the future-proof internet scale assessment of vulnerabilities. His most recent publications include a significant contribution towards making the IPv6 internet scannable, understanding and mitigating the impact of DNS misconfigurations in the DNS ecosystem, and the first study on system operators' perspective on security misconfigurations.

MICHEL VAN EETEN is currently a Professor at the Delft University of Technology and his chair focuses on the Governance of Cybersecurity. His team analyses large-scale internet measurement and incident data to identify how the markets for internet services deal with security risks. He has conducted empirical studies funded by NWO, the ITU, the OECD, the Department of Homeland Security, the European Commission, the Dutch National Police, the General Intelligence and Security Service, Fox-IT, banks, and various ministries. He is also a member of the Cyber Security Council, an Official Advisory Body of the Dutch Government.