On-premises Analysis of Advanced Threat Prevention Appliances

Akira Fujita
National Institute of Information and Communications Technology
Tokyo, Japan
a.fujita@nict.go.jp

Tao Ban
National Institute of Information and Communications Technology
Tokyo, Japan
bantao@nict.go.jp

Takeshi Takahashi
National Institute of Information and Communications Technology
Tokyo, Japan
takeshi_takahashi@nict.go.jp

Daisuke Inoue
National Institute of Information and Communications Technology
Tokyo, Japan
dai@nict.go.jp

ABSTRACT

Cyberattacks are becoming increasingly diverse and serious. To counter them, many organizations run multiple appliances, rather than a singular appliance, because their standard is that there should never be an undetected threat. Although applying those various security appliances has increased organizations' security, their security operations are experiencing alert fatigue, possibly causing incidents due to missing critical threat information or human error. In this research, we investigated how much of the alerts issued by different security devices installed on the same network can be considered duplicates or unique. We obtained the alert data for an organization with multiple appliances for a period of 10 months and extracted all the sets of alerts that could be inferred to refer to the same event to analyze the extent to which the alert types that they generate co-occur across the appliances. According to the analysis of the similarity between alert types based on their co-occurrence, we mapped the alert types in 2 dimensions to discuss the appliances' correlation. We observed that some appliances completely overlap with the alerting behaviors of other appliances and identified appliances that produce many useful alerts with high uniqueness.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**.

KEYWORDS

security appliance, alert fatigue, co-occurrence analysis

CSET '21, August 9, 2021, Virtual, CA, USA

ACM ISBN 978-1-4503-9065-1/21/08…$15.00
https://doi.org/10.1145/3474718.3474720

ACM Reference Format:
Akira Fujita, Tao Ban, Takeshi Takahashi, and Daisuke Inoue. 2021. On-premises Analysis of Advanced Threat Prevention Appliances. In *Cyber Security Experimentation and Test Workshop (CSET '21), August 9, 2021, Virtual, CA, USA*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3474718.3474720

1 INTRODUCTION

Cyberattacks have become increasingly diverse. Accordingly, many organizations had to implement countermeasures [13], for example, installing and operating a security appliance.

Security appliances initially offered a single function such as a firewall, an intrusion detection system (IDS) or antimalware as a response to various threats. The methods, routes, and causes of attacks (e.g., malware) are becoming increasingly complicated, and responding to threats by using single-function defense tools has become impossible. To manage these threats in a 1-step, unified threat management (UTM) [20], a system that detects and responds to various threats with a single unit, is being provided by some developers. Security appliances including those UTMs are being deployed in every organization. Additionally, tools that collect logs of events and analyze those logs in real time, such as security information and event management (SIEM) [2], are now being applied.

However, because of many organizations' concern that there should never be an undetected threat, they run multiple appliances, and the more mission-critical the systems that the organization operates, the more likely it is to engage in that practice.

The appliances output many alerts daily, including relatively unproblematic alerts. As a fact, a single alert may save the organization, for example in context of APTs. Thus, it may not necessarily to be pessimistic in the face of a massive number of alerts. However, it is also a fact that a limited number of security operations staff check those alerts daily to assess whether countermeasures are required. The amount of alerts is not always proportional to the number of security operations staff. Although the application of those various security appliances has increased organizations' security level, their security operations are experiencing alert fatigue [8]. The limited information processing capacity of the security operations staff creates a bottleneck that negatively affects the accuracy and

efficiency of security measures [12]. Many operations staffs are exhausted, increasing the probability of incidents due to missing critical threat information or human error. This problem should be solved.

To address the problem, we investigated how much of the alerts issued by different security devices installed on the same network can be considered duplicates and how much of the alerts can be considered unique. We collected data in the actual network environment used in the business and analyzed how many competing alerts were issued by security appliances. We 1) obtained for 10 months all the alert data for an organization with multiple appliances, 2) extracted all the sets of alerts that can be inferred to refer to the same event, and 3) analyzed the extent to which the alert types that they generate co-occur across the appliances. According to the similarity between alert types on the basis of their co-occurrence, we mapped the alert types in 2 dimensions. As a result of the mapping, several clusters formed. Analysis of these clusters revealed that some appliances completely overlapped with the alerting behaviors of other appliances. We also identified appliances that produce many useful alerts with high uniqueness.

According to our review of the literature, this research is the first to comparatively analyze long-term alert data from multiple appliances installed in a real network.

The main findings of this paper are as follows:

- Alert behaviors of one appliance may completely overlap with behaviors of other appliances.
- On the other hands, one appliance may produce many useful alerts with high uniqueness.

These findings imply that if security analysts eliminate the appliances that do not provide useful information from the group of complementary appliances, the alert processing cost can be reduced and the operation system improved such that additional resources can be spent on appliances with high reliability.

2 DATASET

2.1 Alert data collection method

We collected alert data issued by 14 security appliances (appliane ID: A-N). Those alert data were anonymized to protect the privacy of the network users. The appliances with UTM listed in the function have more than 1 function in web filtering, antimalware, antispam mail, function of IDS/intrusion prevention system (IPS), advanced persistent threat (APT) measures, and firewall. 9 appliances (appliance A, B, F, H, I, J, L, M and N) are UTM. 2 appliances (appliance C and G) are IDS/IPS. Appliance D is an APT measure. Appliance E is an anti-malware. Appliance K is an antispam mail. All the appliances are commercial.

All the aforementioned security appliances were installed into a network of a research institute (the name is blinded for review), to protect the network from unwanted traffic. The network traffic monitored by each appliance is the same. By performing multiple checks on the same traffic in this way, the research institute tried to ensure the detection of incidents and their signs.

The network in which the appliances were installed was assigned a /16 IPv4 global IP address range. More than 1,000 staff members used this network to communicate with external organizations, use external services, and communicate research data.

We collected all the alerts issued by the 14 appliances for 10 months: between Jan 1 and Oct 31, 2017 (304 days). As for Appliances F and H, since those tend to generate a large amount of alerts for minor problems, we set thresholds regarding the level of importance contained in the alerts and excluded minor alerts from the collection. All alerts were aggregated from the appliances and logged on a single server in real time. A total of 137,699,151 alerts were collected.

2.2 Data Field of Alert Data

The structure of the collected alert data is all different for each security appliance. We extracted the following attribute information from all alert data.

Time Date and time when the alert was issued.
Appliance ID ID of the security appliance that issued the alert.
Source IP IPv4 address that sent the communication that the alert was issued.
Destination IP IPv4 address that received the communication that the alert was issued.
Alert type Content of the alert issued by the appliance.

Each attribute's information comprises 1 value, and never more than 1 value in 1 attribute. For the "Time," because time information did not exist in the alert data issued by the appliance in some cases, we used the system time of the server to aggregate and record alerts for all appliances. "source IP" or "destination IP" sometimes had no value, depending on the content of the alert; in such cases, it was recorded as "na" (not available). The content ("alert type") is expressed in words or sentences. There are some limited patterns of content for each appliance. The diversity of the content varies by appliance. Some appliances present a wide variety of alert content to the user, such as referring to the name of the application layer service to indicate the specific type of threat (e.g., "telnet: multiple vendors bsd telnetd encryption key..."); other appliances only convey that the communication is potentially dangerous (e.g., "threat"). Content that differs by at least 1 character is treated as a different alert type.

2.3 Aggregate of Alert Data

Table 1 provides the number of alerts issued by each appliance and the number of alert types.

Of the appliances, B, C, F, G, and I issued the most alerts. Appliances such as A and L had among the least alerts. Variety in the number of alert types and the frequency was observed. Appliance G issued many alerts of many types; by contrast, appliance F issued many alerts of 1 type.

The frequency of alert issuance for all 1,451 alert categories had a mean of 94,900, a median of 30, a maximum of 55,349,924, a minimum of 1, and a sample standard deviation of 1,546,782. Thus, the distribution had a large frequency spread between the high and low frequency alerts. Table 2 shows frequency distribution of the issued times for the 1,451 alert types. A wide variety of very low and high frequency alert types was observed.

Table 1: Breakdown of all alert data

Appliance ID	Amount of alerts	Number of types
A	469	1
B	11,857,488	168
C	66,212,482	437
D	564,561	6
E	6,498	4
F	15,793,731	1
G	36,676,039	811
H	7,253	3
I	6,106,558	11
J	296,972	4
K	89,137	1
L	24	1
M	52,696	2
N	35,243	1
Total	137,699,151	1,451

3 ANALYSIS

3.1 Preliminary Analysis: Extraction of Reminding Alert

In general, most appliances continued issuing other alerts with the same content, unless the problem that triggered the alert was addressed and resolved. The alerts not triggered by an event that first occurred at the time had to be excluded from the co-occurrence analysis of alerts. Because if we counted these alerts in the co-occurrence analysis, co-occurrences of past events would be counted, and the results would not accurately reflect the actual co-occurrence. We referred to these alerts as "reminding alerts" and extracted them in advance to exclude them from co-occurrence analysis.

No data were available to verify which alerts were reminding alerts. However, reminding alerts existed in the alert group in which each alert was issued from the same appliance on the same source IP and destination IP. In addition, we assumed that reminder alerts tend to be set to continue issuing at a certain interval determined for each appliance and alert type. Therefore, the reminding alerts were estimated based on the intervals between alerts.

We extracted alert groups in which all alerts were the same alert type that occurred for the same source IP and destination IP, for each appliance. The alert groups were extracted in 9 appliances. No alert groups were extracted in the remaining 5 appliances (appliances J, K, L, M, and N). We sorted each alert in the alert groups by time value in the alert data in ascending order. Based on the estimation that a reminding alert would probably not be issued more than 1 day later, if there was a time interval of more than 1 day within an alert group, we split the group at that point and divided it into 2 alert groups. We observed interval times between the alerts in the split alert groups.

Figure 1 presented histograms of interval times between the alerts. The histograms for 9 appliances are aligned vertically. Three histograms in a different time window are aligned horizontally for each appliance. From left to right, each histogram demonstrates 1) the frequency distribution of the interval between 0 and 60 seconds, counted in 1-second increments; 2) the interval between 0 and 60 minutes, counted in 1-minute increments; and 3) the interval between 0 and 24 hours, counted in 1-hour increments. The vertical axis in each histogram that presents the frequency of the interval time is represented by a logarithmic scale.

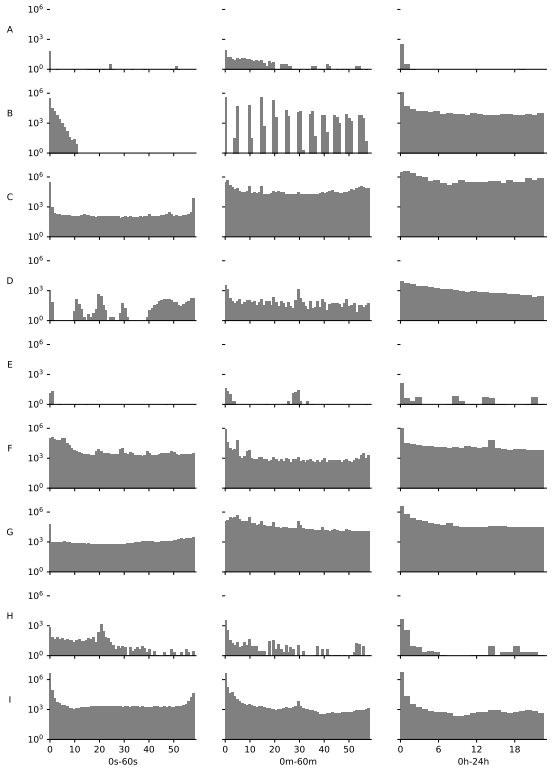

Figure 1: Histograms of Interval Time between Alerts

Most appliances had multiple spike points in the histogram in seconds or minutes. Few appliances had spike points in the histogram in hours. The spike points were considered to represent the time preset as the elapsed time at which the appliance issued a reminding alert. Because if an interval time of reminding alert was fixed, frequency of the time should be discriminately more than that of the other interval times. However, because of the errors in the timing of issuing the alert, the interval times of the reminding alert may be included in the class around the spike point. In addition, depending on the alert type, because the interval settings for reminding alerts might differ for the same appliance, clear spike points might not be observed. Therefore, we used the maximum of the clearly observed spike points (fixed interval values for reminding alerts), and for the intervals below the maximum value, we approximately determined that the alert after that interval was a reminding alert. Specifically, alerts issued with the same alert

Table 2: Frequency distribution of the issued times for each alert type

Range	$10^0 \leq n < 10^1$	$10^1 \leq n < 10^2$	$10^2 \leq n < 10^3$	$10^3 \leq n < 10^4$	$10^4 \leq n$
Number of Alert types	549	344	228	145	185

type for the same source IP and same destination IP after an interval smaller than the threshold time shown in the table 3 were determined to be reminding alerts.

When we extracted the alerts using the condition, the number of alerts presented in table 4 were determined to be reminding alerts. Appliances for which alerts issued with the same alert type for the same source IP and same destination IP were not detected, we assumed that no reminding alert was issued.

3.2 Co-occurrence Analysis

We analyzed how many of the observed alerts were issued for the same event, by alert type. We captured events that occur for which an alert of 1 alert type co-occurs with an alert of another alert type and determined the similarity between the alert types in a round-robin, based on the frequency of co-occurrences. Specifically, we extracted alert pairs that occurred within a certain time interval and had the same set of source IP and destination IP and counted the number of co-occurrences per alert type. On the basis of the number of co-occurrences among all alert types, a similarity matrix among alert types was created, and the matrix was converted into a two-dimensional map by a dimensionality reduction algorithm. The two-dimensional map was used as the object of discussion.

To create the two-dimensional map, we performed a procedure to count the number of co-occurrences between the alert types issued by each appliance and visualize the similarity between alert types, according to the following steps "preprocessing" and "proposed method".

3.2.1 *Preprocessing.*

(1) Collect all the alerts issued by the 14 appliances during the 304 days.
(2) Extract reminding alerts from the collected alerts by using the method presented in chapter 3.1.
(3) Exclude all extracted reminding alerts from the entire collected alerts.
(4) Exclude alerts where the source IP or destination IP is "na" (not available) from the entire collected alerts.
(5) Divide all alerts into alert sets in which the alerts' issue time, source IP, and destination IP match.

3.2.2 *Proposed method.*

(1) Define a time window for all 10 seconds in the observation period (304 days), by shifting the starting point of the window by 1 second from the beginning of the observation period.
(2) Merge alert sets with the same source IP and the same destination IP within the duration for each time window. Discard merged alert sets with only 1 alert.
(3) Extract alert pairs for all combinations in the merged alert set. Consider the extracted pair of alerts to have co-occurred. Add 1 to the number of co-occurrences of the alert type in both alerts each time 1 alert pair co-occurrence is detected. Record the IDs of the alert pair, to control and prevent duplicate retrieval in subsequent attempts.
(4) Calculate the similarity in a round-robin between all alert types and create a matrix: the matrix represents all alert types in both rows and columns, and the component (i, j) represents the similarity between alert type i and alert type j. Calculate a Jaccard coefficient [9] to assess the similarity between alert types: let X be the set of alerts that are of alert type i and Y be the set of alerts that are of alert type j, the Jaccard coefficient between set X and set Y is provided by the equation (1).
$$\frac{|X \cap Y|}{|X \cup Y|} \tag{1}$$
For example, the Jaccard coefficient between alert type i and alert type j was obtained by dividing the number of times the alerts of i and j co-occurred by the number of times the alerts of i or j occurred.
(5) Convert the matrix into a two-dimensional map by a dimensionality reduction algorithm. We applied the t-SNE (t-Distributed Stochastic Neighbor Embedding) algorithm [22] to the dimensionality reduction.

In this manner, using the alert dataset as input, we obtained a two-dimensional scatter plot representing the relative positional relationships of alert types based on the similarity between the alert types for each appliance.

Additionally, 18,043,333 alerts in the observed data contained "na" in the IP address information and were thus excluded in step 4.

4 DISCUSSION

The co-occurrence analysis resulted in the two-dimensional map (scatter plot) of the relative positions between alert types presented in figure 2.

Each marker in the scatter plot represents an alert type. The color of the markers represents the appliance to which the alert type belongs. The size of the markers represents the frequency of the alert type issued. The diameter of the marker is proportional to the normal logarithm of the frequency. When the result of calculating the normal logarithm was below 1, we set the same diameter as when the result was 1, for visibility. The scale of the coordinates is determined by the results of the t-SNE calculation. The unit of the marker diameter is independed to the scale of the coordinates. The dashed circle in figure 2 indicates the boundary between alert types that appears to be cohesive and a group of alert types are distant from other alert types. We referred to the alert types outside the boundary as "isolated alert types." The dashed circle that represents the boundary was drawn at the discretion of the authors.

There is a concentration of markers in the center of the right half of the figure 2, and in the same figure, several clusters of particularly

Table 3: Threshold for Interval Time to Determine Reminding Alert

Appliance ID	A	B	C	D	E	F	G	H	I
Threshold Time	0s	55m	15m	30m	30m	15h	30m	21s	30m

Table 4: Number of Extracted Reminding Alerts

Appliance ID	A	B	C	D	E	F	G	H	I
# of Reminding Alert	66	1,215,712	1,646,873	7,871	109	1,314,600	3,492,486	3,242	4,749,620

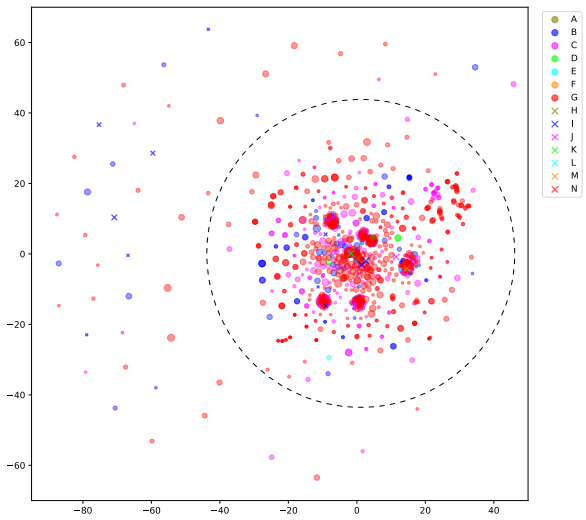

Figure 2: Relative Positional Relationships of alert types

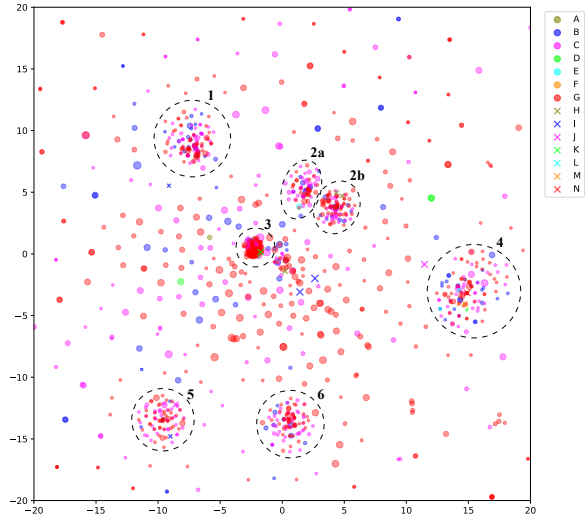

Figure 3: Relative Positional Relationships of Alert Types: Enlarged View around the Clusters

aggregated markers around the rectangular are from [-20,-20] to [20, 20]. Figure 3 presents the enlarged view around the area.

According to figure 3, we identified 7 clusters. We referred to the clusters as "Cluster_1, 2a, 2b, 3, 4, 5, 6 and 7" for each, as presented as the dashed circles in figure 3. The alert types contained in the 7 clusters were highly cohesive. The inclusion of an alert type in these clusters might indicate that the alert type had an especially high probability of responding to the same event as the other alert types in the cluster. In other words, for pairs of alert types with different marker colors in the cluster, the appliance tended to issue alerts that duplicated those of the other appliance. A pair of alert types that had the same marker color in the cluster indicated that for a single event, there was a situation for which the same appliance issued multiple alerts with different content related to the event; alternatively, it indicated that for 1 event, there was often an event that was likely to occur in conjunction with it.

4.1 Analysis of Clusters

We observed the alert types within the 7 identified clusters. The number of alert types in the clusters for each appliance is presented

Table 5: Breakdown of Clusters

Cluster ID	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Total Number of Included alert types
1	0	22	58	0	0	0	48	0	0	0	0	0	0	0	128
2a	0	14	41	0	1	0	46	0	0	0	0	0	0	0	102
2b	0	13	33	0	1	0	48	2	0	0	0	0	0	0	97
3	0	10	32	2	0	0	75	0	0	1	1	0	0	0	121
4	0	24	43	1	1	0	56	0	0	0	0	1	2	1	129
5	0	4	37	0	0	0	46	0	1	1	0	0	0	0	89
6	0	10	48	0	0	0	47	0	0	1	0	0	0	0	106

in table 5, and many of the alert types of appliances B, C, and G are included in the clusters. However, because all 3 appliances have relatively more alerts than the others do in the number of alert types and the number of alerts issued, the 3 appliances probably do not have characteristics that would make them tend to duplicate the alerts issued by other appliances.

To analyze which appliances issued alert types that would probably overlap with other appliances, we used the formula (2) to determine how many of the total alert types for that appliance would probably be duplicated by other appliances. The value from

the formula 2 represented how likely the type of alert issued by appliance m was to be encompassed by the behavior of another specific appliance n.

$$InclusionRate(m,n) = \frac{\sum_{c \in Clusters} n_c^m * b_c^n}{N_m} \quad (2)$$

In equation (2), N_m is a total number of alert types by appliance m, n_c^m is a number of alert types by appliance m in a cluster c, and b_c^n is an existence of alert type by appliance n in cluster c. When there is 1 or more alert type by appliance n in cluster c, the b_c^n will be "1," and "0" otherwise.

Table 6 presents the result of the inclusion rate calculation. According to table 6, appliances K, L, M, and N had the inclusion rate of 1.00 to some other appliances. These appliances were completely covered by other appliances in terms of the alerts they issued and do not need to be added in an environment in which particular appliances are applied to. For appliances B, C, D, and E, more than half of the alert types were encompassed in the other particular appliance. Depending on the purpose for which the appliance is being applied, the choice of not deploying it may be considered. Additionally, appliances A, F, and I were independent of other appliances because their inclusion rates were 0.00 for the most part, namely, appliances F and I issued many alerts, according to table 1, and in terms of the coverage of events that occur in network, those appliances are worth keeping in the environment.

We analyzed what type of set of alerts each of the clusters that were formed was, to understand what topics of events for which duplicate alerts would probably be issued. We considered the set of alert content in the alert types for each cluster to be a single document, calculated the TF-IDF values [15][10] of the occurring words, and listed the words with TF-IDF values up to the top 10 in table 7. To calculate TF-IDF, we used the functions contained in the python Scikit-learn library [17]. The max_df value was set to 0.9. Symbols () :′ ” in the alert type were deleted. Spaces, hyphen, and underscore were used as delimiters of words.

Cluster_1 seems to indicate the detection of outbound communication with CNC servers by trojan-type malware on Windows platforms. Cluster_2a seems to indicate the detection of a potentially unwanted application of the adware-type targeting Windows. In cluster_2b, alerts on integer overflow in web-browsing images or graphics seem to be featured; however, overall, there is a complex of alerts for various categories of web browsing. Both clusters_2a and _2b can be considered the set of alerts related to web browsing. We assumed is this finding is why both clusters are mapped close to each other. The cluster_3 seems to be a set of alerts issued in conjunction with or related to a port scan by a bot. The alert types in cluster _4 seem to be related to trojan emails. Cluster _5 appears to be a set of alerts for a vulnerability related to loading dynamic link libraries; however, it is a complex cluster that includes alert types unrelated to that. Cluster_6 is considered to be a set of alerts related to SQL injection.

4.2 Isolated Alert Types

Because the isolated alert types were rarely simultaneously issued with alerts from other alert types, those are highly unique and relatively valuable. However, this alert type could also be an alert type that tends to be issued at a time different from the time of issuance of other appliances, due to false event detection.

The isolated alert type comprised 47 types: 9 for appliance B, 5 for C, 27 for G, and 6 for I. Many of the types belonged to appliance G. An infrequent alert type tended to be an isolated alert type because it was issued less frequently than the other types and had a lower probability than the other types of being issued simultaneously with other alerts. Notably, frequently issued isolated alert types may be highly unique alerts if the contents differ from that of alert types in the clusters; otherwise, false event detection may occur.

According to the frequency of the 47 issued alert types, those with a high frequency, namely, over 100,000 times, were the 3 types issued by appliance G: 1) alert for invalid DNS flows, 2) alert that the data field of the NTP control message is too long, and 3) notification of a handshake with a transferring BitTorrent file. These 3 alert types differ from those in the clusters, and depending on the trend of the attack, they may occur more frequently at times. At least those 3 types seemed not to be alerts by false event detection, and they are sufficiently useful.

5 RELATED WORK

Many studies have assessed computer network protection techniques to evaluate the performance of individual techniques. A particular area of study for research institutes has been evaluating IDS systems. In paper [1], the author tested and analyzed the performance of the IDS systems Snort [18] and Suricata [6], general open source IDS systems. In paper [21], the author investigated the performance and detection accuracy of Snort, Suricata, and Bro [16] and IDS systems, using various attack types, including DoS attacks, DNS attacks, FTP attacks, scan port attacks, and SNMP attacks. The author found that the detection accuracy decreases under certain conditions. In paper [23], the author stated that the intrusion detection methods proposed in the literature have not been reliably evaluated for real-world use. According to that statement, the author proposed a new evaluation method for the field of machine learning intrusion detection. Studies have also evaluated other protection functions. In paper [19], the author reviewed the datasets to test existing automated APT detection methods. In paper [3], the author reviewed of content-based e-mail spam filtering techniques.

There are some studies that compare appliances across the board and examine how they work. In paper [4], the author evaluated the services of two commercial threat intelligences as to what these services consist and compare their metrics, and compared with four large open threat intelligence feeds. In paper [7], the author assessed the quality of 17 open source cyber threat intelligence feeds for over a year. In paper [14], the author defined a set of metrics to characterize threat intelligence data feeds and characterize public and commercial sources. In paper [5], in order to know if a security appliance is what cybersecurity analysts should install and if it is necessary for their network before they install it, using data from the Managed Security Service Providers (MSSP) data, the author developed a virtual security appliance in an attempt to predict incidents that would have been detected if the appliance had been present. In paper [11], the author tried to enable the aggregation of related alerts generated by a single security appliance.

Table 6: Inclusion Rate of Alert Types

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	-	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
B	.00	-	.58	.20	.30	.00	.58	.08	.02	.14	.06	.14	.14	.14
C	.00	.67	-	.17	.27	.00	.67	.08	.08	.27	.07	.10	.10	.10
D	.00	.50	.50	-	.17	.00	.50	.00	.00	.33	.33	.17	.17	.17
E	.00	.75	.75	.25	-	.00	.75	.25	.00	.00	.00	.25	.25	.25
F	.00	.00	.00	.00	.00	-	.00	.00	.00	.00	.00	.00	.00	.00
G	.00	.45	.45	.16	.18	.00	-	.06	.06	.21	.09	.07	.07	.07
H	.00	.67	.67	.00	.67	.00	.67	-	.00	.00	.00	.00	.00	.00
I	.00	.09	.09	.00	.00	.00	.09	.00	-	.09	.00	.00	.00	.00
J	.00	.75	.75	.25	.00	.00	.75	.00	.25	-	.25	.00	.00	.00
K	.00	1.00	1.00	1.00	.00	.00	1.00	.00	.00	1.00	-	.00	.00	.00
L	.00	1.00	1.00	1.00	1.00	.00	1.00	.00	.00	.00	.00	-	1.00	1.00
M	.00	1.00	1.00	1.00	1.00	.00	1.00	.00	.00	.00	.00	1.00	-	1.00
N	.00	1.00	1.00	1.00	1.00	.00	1.00	.00	.00	.00	.00	1.00	1.00	-

Table 7: Top 10 characteristic words in each cluster (TF-IDF value)

Cluster ID	Words
1	cnc(.24) trojan(.24) win(.19) excel(.17) outbound(.17) known(.15) snmp(.13) blacklist(.13) assertion(.11) failure(.11)
2a	response(.16) pua(.16) executable(.13) known(.13) win(.13) adware(.12) ms(.12) mytransitguide(.11) sandbox(.11) activex(.11)
2b	firefox(.15) integer(.14) archive(.12) too(.12) jpeg(.12) kerberos(.12) script(.12) self(.12) signed(.12) technology(.12)
3	portscan(.27) invalid(.19) smtp(.19) flow(.18) tcp(.17) bot(.15) long(.14) confidence(.13) high(.13) portmapper(.13)
4	email(.30) mal(.22) trojan(.20) database(.19) cnc(.18) smtp(.18) reputation(.15) dangerous(.15) domain(.14) win(.12)
5	dll(.17) download(.14) os(.14) trojan(.13) iax2(.12) option(.12) peer(.12) potentially(.12) rtsp(.12) vlc(.12)
6	sql(.20) download(.17) 2017(.15) get(.13) rdp(.13) rtf(.13) wordpress(.13) generic(.12) init(.11) method(.11) netcat(.11)

However, despite such studies that evaluate the detection performance of individual functions in isolation or some threat intelligences (security appliances), our research is the first to obtain and analyze actual operational data on the synergistic effect and the redundancy when multiple appliances are configured simultaneously.

6 ETHICAL CONSIDERATION

The data obtained in this study were the log of alerts from security appliances installed in a network actually used for business purposes. All the data were not fictitious but were alerts issued for communications made by a specific person or for the configuration of a device actually in operation. It is inappropriate for analysts to be able to identify communications made by individuals (e.g., source address/destination address/content of communication) or which device was actually the target of the alert, by analyzing the data. Therefore, in our research, the source IP and destination IP of each alert data included in the data to be handled were hashed respectively in advance so that the actual IP address could not be identified by the analyst. In addition, to ensure that only analysts could view the alert data, the data was recorded on a server placed in a physically locked room, and a permission setting and authentication process was established to ensure that only analysts could access the data.

7 CONCLUSION

In this paper, we introduced the results of the investigation how much of the alerts issued by different security devices installed on the same network can be considered duplicates or unique. We obtained the alert data for an organization with multiple appliances for a period of 10 months and extracted all the sets of alerts that could be inferred to refer to the same event to analyze the extent to which the alert types they generated co-occurred across the appliances. According to the analysis of the similarity between alert types on the basis of their co-occurrence, we mapped the alert types in 2 dimensions to discuss the appliances' correlation.

In conclusion, we demonstrated that some appliances completely overlap with the alerting behaviors of other appliances and identified appliances that produce many useful alerts with high uniqueness. Further experiments are required to examine that by dropping the operation of those overlapping appliances and devoting resources to useful appliances, the network will continue to be secure, and the load on security operators will be reduced.

ACKNOWLEDGMENTS

This research was conducted under a contract of "MITIGATE" among "Research and Development for Expansion of Radio Wave Resources(JPJ000254)", which was supported by the Ministry of Internal Affairs and Communications, Japan.

REFERENCES

[1] Adeeb Alhomoud, Rashid Munir, Jules Pagna Disso, Irfan Awan, and A. Al-Dhelaan. 2011. Performance Evaluation Study of Intrusion Detection Systems. 5 (2011), 173–180.

[2] Mark Nicolett Amrit T. Williams. 2005. Improve IT Security with Vulnerability Management.

[3] Alexy Bhowmick and Shyamanta M. Hazarika. 2018. E-Mail Spam Filtering: A Review of Techniques and Trends. (2018), 583–590.

[4] Xander Bouwman, Harm Griffioen, Jelle Egbers, Christian Doerr, Bram Klievink, and Michel van Eeten. 2020. A different cup of TI? The added value of commercial threat intelligence. In *Proceedings of 29th USENIX Security Symposium (USENIX Security 20)*. 433–450.

[5] Shang-Tse Chen, Yufei Han, Duen Horng Chau, Christopher Gates, Michael Hart, and Kevin A. Roundy. 2017. Predicting Cyber Threats with Virtual Security Products. In *Proceedings of the 33rd Annual Computer Security Applications Conference*. 189–199.

[6] The Open Information Security Foundation. 2010. Suricata Open Source IDS/IPS/NSM engine. https://suricata-ids.org/.

[7] Harm Griffioen, Tim Booij, and Christian Doerr. 2020. Quality Evaluation of Cyber Threat Intelligence Feeds. (2020), 277–296.

[8] Wajih Ul Hassan, Shengjian Guo, Ding Li, Zhengzhang Chen, Kangkook Jee, Zhichun Li, and Adam Bates. 2019. NODOZE: Combatting Threat Alert Fatigue with Automated Provenance Triage. In *Proceedings of Network and Distributed Systems Security (NDSS) Symposium 2019*.

[9] Paul Jaccard. 1912. The Distribution of the Flora in the Alpine Zone.1. *New Phytologist* 11, 2 (1912), 37–50.

[10] Karen Spärck Jones. 1972. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation* 28 (1972), 11–21.

[11] Klaus Julisch. 2003. *Using root cause analysis to handle intrusion detection alarms*. Ph.D. Dissertation. Technical University of Dortmund, Germany.

[12] Faris Bugra Kokulu, Ananta Soneji, Tiffany Bao, Yan Shoshitaishvili, Ziming Zhao, Adam Doupé, and GailJoon Ahn. 2019. Matched and Mismatched SOCs: A Qualitative Study on Security Operations Center Issues. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1955–1970.

[13] Platon Kotzias, Leyla Bilge, Pierre-Antoine Vervier, and Juan Caballero. 2019. Mind Your Own Business: A Longitudinal Study of Threats and Vulnerabilities in Enterprises. In *Proceedings of Network and Distributed Systems Security (NDSS) Symposium 2019*.

[14] Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, Stefan Savage, and Kirill Levchenko. 2019. Reading the Tea leaves: A Comparative Analysis of Threat Intelligence. In *Proceedings of 28th USENIX Security Symposium (USENIX Security 19)*. 851–867.

[15] H. P. Luhn. 1957. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development* 1, 4 (1957), 309–317.

[16] Vern Paxson. 1998. Bro: A System for Detecting Network Intruders in Real-Time. In *Proceedings of the 7th Conference on USENIX Security Symposium*.

[17] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[18] Martin Roesch. 1999. Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of the 13th USENIX Conference on System*. 229–238.

[19] Branka Stojanović, Katharina Hofer-Schmitz, and Ulrike Kleb. 2020. APT datasets and attack modeling for automated detection methods: A review. 92 (2020).

[20] Kenneth Tam, Martin, Hoz Salvador, Ken McAlpine, Rick Basile, Bruce Matsugu, and Josh More. 2013. *UTM Security with Fortinet*. Syngress.

[21] Kittikhun Thongkanchorn, Sudsanguan Ngamsuriyaroj, and Vasaka Visoottiviseth. 2013. Evaluation studies of 3 intrusion detection systems under various attacks and rule sets. In *Proceedings of 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*. 1–4.

[22] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data Using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.

[23] Eduardo K. Viegas, Altair O. Santin, and Luiz S. Oliveira. 2017. Toward a reliable anomaly-based intrusion detection in real-world environments. 127 (2017), 200–216.