

HOLMES: An Efficient and Lightweight Semantic Based Anomalous Email Detector

Peilun Wu* and Hui Guo[§]

Data Security & Compliance, CDO Data & Cloud, PwC CN.*

School of Computer Science and Engineering, University of New South Wales (UNSW)*[§]

Email: *z5100023@zmail.unsw.edu.au, [§]h.guo@unsw.edu.au

Abstract—Email threat is a serious issue for enterprise security. The threat can be in various malicious forms, such as phishing, fraud, blackmail and malvertisement. The traditional anti-spam gateway often maintains a greylist to filter out unexpected emails based on suspicious vocabularies present in the email’s subject and contents. However, this type of signature-based approach cannot effectively discover novel and unknown suspicious emails that utilize various evolving malicious payloads. To address the problem, in this paper, we present HOLMES, an efficient and lightweight semantic based engine for anomalous email detection. HOLMES can convert each email event log into a sentence through word embedding and then identify abnormalities that deviate from a historical baseline based on those translated sentences. We have evaluated the performance of HOLMES in a real-world enterprise environment, where around 5,000 emails are sent/received each day. In our experiments, HOLMES shows a high capability to detect email threats, especially those that cannot be handled by the enterprise anti-spam gateway. It is also demonstrated through our experiment that HOLMES can discover more concealed malicious emails that are immune from several commercial detection tools.

Index Terms—phishing detection, novelty detection, machine learning, intrusion detection, fraud detection.

I. INTRODUCTION

Though the instant messaging software, such as Facebook and WeChat, has gained increasing popularity, the email service is still indispensable for enterprises. Since the email service is a public-facing application, it can be targeted by the hacker as an easy entrance to the internal network. Based on our observations, fraud, malvertisement and spread-phishing are the main email threats frequently received by enterprise users. These emails use deceptive subjects to pretend and hide themselves. Usually, malware infected attachments or malicious URLs are embedded in the email body to spoof recipients for further action. Once the attachment is downloaded or a link is clicked, the recipients’s system is compromised or the confidential information is leaked [1].

To alleviate the problem, an enterprise often deploys some anti-spam gateways to filter out unexpected emails. However, the associated techniques for spam detection, such as greylist and subject analysis, cannot effectively discover novel and unknown email threats that are elaborately constructed by utilizing various current hot topics, such as COVID-19, US election. These unknown threats can easily

bypass the anti-spam gateway and successfully permeate the target system, leading to a series of damaging consequences, such as administrator account theft, database attack and financial blackmail.

In this paper, we introduce a novel artificial intelligence based anomalous email detector, HOLMES, that can effectively tackle the challenges mentioned above. HOLMES combines word embedding with novelty detection to discover anomalous behaviours from a high volume of mirrored SMTP traffic in a large-scale enterprise environment. To improve the result interpret-ability, we trace the real source IP addresses of suspicious emails in line with their geographical positions and further visualize the correlated relations in a directed-force graph. Our contributions are summarized as follows:

- We propose an efficient and lightweight semantic based anomalous email detector, HOLMES. Different from other detectors that usually require to examine email bodies, HOLMES can discover anomalies simply based on email headers, which significantly reduces the cost of resource consumption and avoids accessing email bodies (a sensitive security issue).
- We exploit graph visualization to reveal the correlated relations of detected suspicious emails and demonstrate that the attacker portrait (based on their geographical positions) is in line with the cyber threat intelligence provided.
- We evaluate HOLMES with a commercial anti-spam gateway deployed in a real-world enterprise environment. HOLMES not only can accurately detect those email threats that have been blocked by the anti-spam gateway, but also can discover a large number of email threats that have successfully escaped from the gateway. We also compare HOLMES with several commercial email detectors offered by different security vendors in VirusTotal [2], which shows that HOLMES outperforms those detectors with a very high detection rate on the use of threat hunting in the wild.

The remainder of the paper is structured as follows. We begin with a brief discussion of some related work on email detection in Section II. We then in Section III introduce the proposed semantic based anomalous email detector, HOLMES. In Section IV, we present our evaluation results

of HOLMES and several commercial security products; a demonstration of how visualization can be used to reconstruct the attack stories is also given in this section. The enhancements on HOLMES for the real world implementation is given in Section V. The paper is concluded in Section VI. As an add-on section, we append some extra discussions at the end of this paper.

II. PRELIMINARY KNOWLEDGE

Anomalous emails can be classified into external threats and internal threats in accordance with MITRE ATT&CK Matrix [3]. External threats are the emails sent from external sources, whereas the internal threats are the emails sent from legitimate users within an organization but whose email accounts have been stolen and used for the lateral movement attack. Most of previous research mainly focuses on one specific threat type, such as URL-based lateral phishing [4] or phishing web pages from search engine in a large-scale cyberspace [5]. There are still many open questions and unsolved challenges that need to be addressed holistically. Some issues and the existing solutions are presented below.

No Built-In Authentication in SMTP. The lack of a native authentication mechanism inside the SMTP service presents a security loophole to attackers. Attackers can easily forge the email header by pretending to be someone the recipient knows or from a business the recipient has a relationship with, so as to spoof recipients and avoid spam block lists [6]. To address the problem, several frameworks, such as SPF [7] (Sender Policy Framework), DKIM [8] (Domain Key Identified Mail) and DMARC [9] (Domain-Based Message Authentication, Reporting, and Conformance), have been developed to incorporate authentication into the email system. However, these designs are still not very effective in terms of implementation. When integrating authentication into the mail system with a typical component-based software design, there are inconsistency issues between the software components offered by different parties [10], such as the incompatibility of mail forwarding servers, which allows numerous email threats escape detection.

Lack of Sensitivity to Unknown Variations. The unreliability of SMTP leaves email threats to have evolved into many variations, which are difficult to be discovered by the traditional security products. We have evaluated several malicious email detection modules within our internal security products that use pattern matching of attack signatures for anomaly detection. None of them can discover the crafted phishing emails that utilize business-related content to pretend themselves look normal for evasion. We also have used the crafted phishing samples collected from our real-world hunting to evaluate the detection rate of 60+ typical detection engines in VirusTotal (Enterprise Service). Nevertheless,

the evaluation result also shows their low sensitivity to unknown threats – in fact, all testing samples can successfully escape from the detection of those engines. This kind of low ability of detecting unknown attack variations has motivated the security community to turn to AI-based methods for anomaly detection.

High False Positive Rate. The research on anomaly detection for cyber threat hunting has been around for decades. The main concern on applying machine learning for anomaly detection is the significant false positive rate (FPR). Even though new designs are continuously proposed aiming for improvement [11]–[13], they were rarely evaluated in a real-world working environment, let alone put into use in commercial systems.

High Cost and Performance Bottleneck. The imbalance between the cost of data collection and the performance of algorithmic consumption is a significant challenge for most of the AI-based detectors. Though the complexity of AI computing algorithms has been constantly improved, most AI modules still require large computing and storage resources, which makes the existing attack detectors not easy to use and very slow to response attacks. Furthermore, the detectors that use supervised machine learning require the labeled input data records and often need to be retrained once their performance begins to degrade, which also makes the machine learning ineffective for detection automation.

Lack of Provenance Analysis. So far few detectors have considered to integrate the provenance analysis within the detection mechanism. We believe provenance analysis is an important and enabling component in malicious email detection. Provenance analysis [14] can reveal the attack story and the detail of attacker portrait behind the email, such as (1) where the email is from, (2) who the real sender is, (3) how the malicious shellcode executes, (4) what the potential correlations between malicious events are. The above information is important for the security team to analyze the attack techniques, tactics and procedures (TTPs) and further assist the security experts to identify individual attackers or organizations.

III. HOLMES - ANOMALOUS EMAIL DETECTOR

To address the above challenges, we introduce an efficient and lightweight semantic oriented anomalous email detector, HOLMES, that can detect an email attack by analyzing the sender-recipient relation, which is available in the email header of SMTP.

HOLMES is a threat hunting tool for the incident response and investigation. It works on mirrored network traffic to inspect and report anomalies in the bypass but do not block them directly. HOLMES is used to assist the incident response team to discover more concealed threats that can

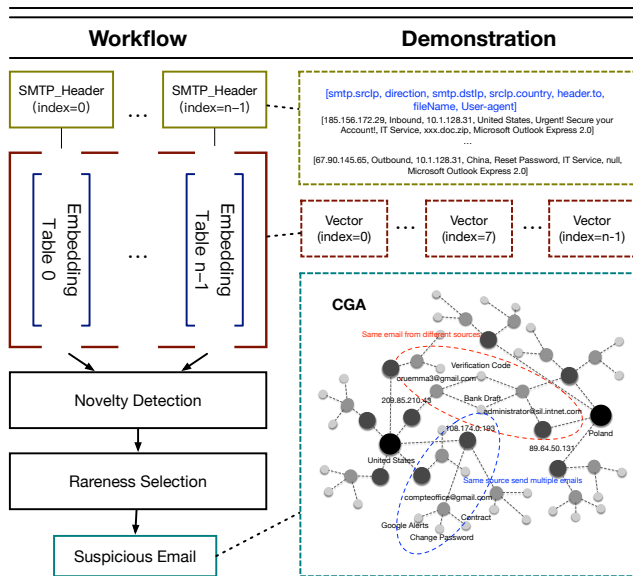


Fig. 1. HOLMES' workflow and demonstration

escape from the detection of traditional anti-spam gateway and threaten the network security.

HOLMES is a self-adaptive learning machine. It can learn historical SMTP traffic from last 24 hours and detect anomalies that deviate from the baseline of historical behaviour in the next 24 hours. Fig 1 shows the overall HOLMES workflow along with the demonstrations as how data/information is evolved in the system. HOLMES consists of four main functions: 1) Word Embedding, 2) Novelty Detection, 3) Rareness Selection, and 4) Correlation Graph Analysis. For each email, HOLMES takes its header and converts it into a numeric presentation through word embedding. The numeric data records are then input to the machine learning unit (Novelty Detection). The unit generates a list of novel emails, which are, in turn, processed by the rareness selection procedure to narrow down the detection targets. The detected results are finally presented in a human readable format and the correlations of the related email attacks are also pictured with a graph.

HOLMES is also a super lightweight and high-efficient detector, which was originally written in Python with only 52 lines of code. Based on the run-time analysis, HOLMES can complete the entire detection in less than 73 seconds with 127 MB memory consumption on around 700 MB datasets. In this section, we would open-source the original code and detail the implementation of the main detection functions in a hope to assist researchers to evaluate and reuse HOLMES in their future research.

A. Word Embedding

Since the email textual header information cannot be directly used for machine learning, **how to effectively represent textual data to the machine understandable** is important. Most algorithm engineers use OneHotEncoder

[15] or OrdinalEncoder [16], or Bag-of-Words (BOW) [17], which can be simply implemented by the open-sourced library Scikit-Learn. However, those methods are not able to effectively maintain the data semantic correlations in either temporal or in spatial dimension.

To address the problem, we use paragraph vector (Doc2Vec) [18] for the conversion, as detailed by the Python code in Listing 1.

```

1 def Doc2Vec(self, feature):
2     """
3     :description: convert textual SMTP header
4     to vectors
5     :param feature: SMTP features
6     :return: word vectors
7     """
8     # build a vocabulary for each feature
9     documents = [TaggedDocument(doc, [i]) for
10 i, doc in enumerate(feature)]
11     # set vector size
12     model = Doc2Vec(vector_size=40, min_count
13 =2, epochs=40)
14     # build model
15     model.build_vocab(documents)
16     # train the model
17     model.train(documents, total_examples=
18 model.corpus_count, epochs=model.epochs)
19     # return vectors
20     return model.docvecs.vectors_docs

```

Listing 1. Doc2Vec

Compared to other conversion methods, Doc2Vec is able to better keep the semantics of the words or more formally the distances between the words, which can be of variable-length ranging from sentences to documents. Doc2Vec is a semi-supervised learning algorithm. Its input is unlabeled but what will be learned is specified/supervised. In our code, the inputs are email headers and what to be learned are the features in the header, as highlighted in blue in the top-right block in Fig. 1. Besides some basic attributes such as subject, header.from or user-agent, which are often forged by hackers, we design two additional features that can also be used to help identify anomalies: the direction of email (direction) and the country of source IP address (srcIp.country). The code in Listing 1 converts each email event to a feature vector, as illustrated in the right side middle boxes in Fig. 1. The feature vectors are then used for novelty detection.

B. Novelty Detection

Anomalous emails are usually unknown and novel. Their behaviors often deviate from the trace of historical normal activities. We use Local Outlier Factor (LOF) [19] to discover those emails, as given in Listing 2 and Listing 3.

```

1 def local_outlier_factor(self, train_feature,
2 test_feature):
3     """
4     :description: estimate the decision score
5     for each event
6     :param train_feature: training data
7     :param test_feature: testing data
8     :return decision scores

```

```

7      '''
8      # initialize the LOF parameters (not
sensitive)
9      LOF = LocalOutlierFactor(n_neighbors=20,
novelty=True, contamination=0.5)
10     # fit training data
11     LOF.fit(train_feature)
12     # return a list of decision scores of the
testing data
13     return list(LOF.decision_scores(test_feature))
on_function(test_feature)

```

Listing 2. Local Outlier Factor (LOF)

The LOF algorithm shown in Listing 2 can learn the feature vectors of historical emails (i.e. the train_feature dataset in the code) then provide the outlier score for newly seen emails (from the test_feature dataset).

There are some compelling advantages of applying LOF for novelty detection: (1) It allows to train learning model on the data with contamination; (2) It has a low computing complexity and can be used for online-learning, hence avoiding the performance degradation and the cost of retraining; (3) It is not sensitive to fine-tuning, which is beneficial to the effectiveness and stability of parametric learning.

```

1  def novelty_analysis(self, factor, test_feature)
:
2      '''
3      :description: filter novel events in
accordance with the decision scores
4      :param factor: decision scores of LOF
5      :param test_feature: testing data
6      :return novel/unseen samples
7      '''
8      # initialize threshold as zero
9      threshold = 0
10     # initialize an empty list to store the
index of outliers
11     outliers = []
12     # initialize an empty list to store
novelties
13     novelties = []
14     # if negative values return the index of
outliers
15     for score in factor:
16         if score < threshold:
17             outliers.append(factor.index(score))
18     # return novelties according to the index
of outliers
19     for index in outliers:
20         novelties.append(test_feature[index])
21     return novelties

```

Listing 3. Novelty Analysis

The decision scores from the LOF code can be negative and positive. The negative values indicate the abnormalities and the positive values indicate the normal behaviours. We regard any vector with a score smaller than a threshold is associated with an anomalous email, which is traced by the its index in the dataset, as shown in Listing 3.

C. Rareness Selection

If we consider the relation of sender and recipient in emails, anomalous emails are often associated with a weak

relation in that **a hacker usually does not send and reply to an email with the same recipient regularly**. We therefore can further narrow down the malicious emails based on the sender-recipient relation – namely, those abnormal emails with a weak sender-receiver relation will be selected as the final detection result, as described in Listing 4.

```

1  def rareness_selection(self, focus_data):
2      '''
3      :description: filter rare events from
novel emails
4      :param data: unseen samples
5      :return rare emails
6      '''
7      # initialize an empty dict to store the
relation
8      relation = {}
9      # initialize an empty list to store the
rareness events
10     rareness = []
11     # initialize the features of a relation:
12     for each_data in focus_data:
13         src_ip = each_data[0]
14         mail_from = each_data[1]
15         mail_to = each_data[2]
16         direction = each_data[3]
17         # define the features of a relation
18         each_relation = src_ip + direction +
mail_from + mail_to
19         # count the relation
20         relation.setdefault(each_relation, [])
.append(each_data)
21     # filter the rare events that are smaller
than a fine-tuned threshold from the relation
dict
22     threshold = 2
23     for k, v in relation.items():
24         if len(v) <= threshold:
25             rareness.extend(v)

```

Listing 4. Rareness Selection

In our design, the relationship is measured based on the combination of a set of email features: source IP (src_ip), direction, sender (mail_from), and receiver (mail_to). For a strong sender-receiver relation, there should be many emails of the same IP-direction-sender-receiver value. Therefore, we count emails for different IP-direction-sender-receiver values and select those that have a low count value (smaller than a threshold) as malicious emails, where the threshold is often fine-tuned based on different network environment by the operation team.

D. Correlation Graph Analysis (CGA)

Most of prior research overlooked a problem: **what is the relation within the anomalies?** Lack of an effective solution significantly increases the load of security analysts, blurs the attacker portraits, and further makes the provenance analysis difficult. To address the problem, we introduce a correlation graph analysis (CGA) module to improve the clarity of attacker portrait descriptions by correlating different anomalous events.

CGA is a directed-force graph and in our design, each node consists of the selected header features: country, srcIp, sender and subject. The directed graph enforces the nodes

that have dense connections come closer but separates the nodes if they do not or have sparse connections. The graph depicts the similarity of different anomalies (such as the same srcIp, same subject or same sender) and centralizes the cluster in line with their geographical locations, hence significantly improving the interpret-ability of provenance analysis.

The graph on the bottom right of Fig. 1 demonstrates the visualization result of CGA, where two clusters (one in red and one in blue) highlight the connected components that are centralized in accordance with the country of srcIp. The blue cluster shows that the same malicious email but sent from different sources, and the red cluster reveals the same source sends multiple different malicious emails.

The CGA module can be used to generate active IOCs (Indicator of Compromise) for the Cyber Threat Intelligence Platform, where we can match the similar or same malicious incidents occurred to other customers based on the IOCs.

IV. EVALUATION

HOLMES has been deployed in an enterprise environment, where it can read mirrored SMTP records from the Elastic-Search (ES) server. In this section, we first present some case studies on the malicious emails detected by HOLMES and then show the comparison result HOLMES with other popular commercial email detectors.

A. Case Studies

According to our monthly email system data, HOLMES can discover around 1,000 anomalous emails each day. Among them, about 23% are truly malicious. And most of the malicious emails contain either phishing links or malware infected attachments. The rest are mainly spams and only a few are false positives.

Based on the detection results, we derive some malicious emails from our email server, which were not blocked by the anti-spam gateway but have been identified by our security analysts as the high risks, to reconstruct the attack stories. Here, we would showcase some delicate crafted phishing emails and describe their malicious behaviour in detail.

Case A: Fake DHL Delivery Message: Fig. 2(a) shows the execution flow of a malicious email that pretends DHL service and plays following tricks: (1) The email uses a normally-seen subject that is associated with an invoice document; (2) The sender information has been modified as ‘DHL Express’, which can be implemented by some hacking tools, such as swaks [20] or cobalt strike [21]; (3) The email includes an attachment named *invoice.doc*, which is, in fact, a malicious Trojan document that utilizes the CVE-2017-11882 [22] vulnerability; (4) The email contains a dedicated picture of DHL delivery service to spoof recipients.

In this attack scenario, an attacker who successfully exploited the vulnerability could run arbitrary code in the context of the current user (recipient). If the user is logged on with the administrative user rights, the attacker could

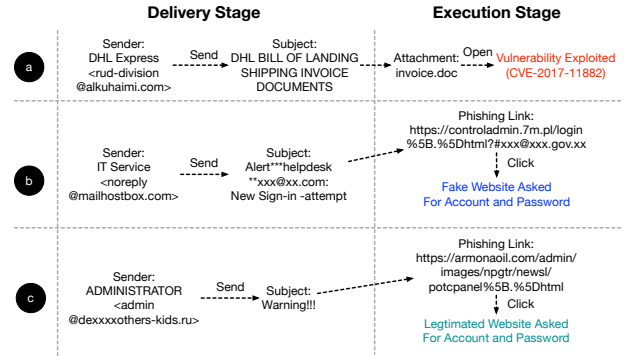


Fig. 2. Case Studies

take control of the affected system. The attacker could then install programs; view, change, or delete data; or create new accounts with full user rights. As we can see, users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights.

Case B: Fake IT Security Alert: Fig. 2(b) shows the execution flow of a malicious email that uses a deceptive subject named “New Sign-in Attempt”, aiming to spoof recipients to change their email account password. Once the recipient clicks the button of “Update security settings”, the web page will be redirected to the phishing website: [https://controladmin.7m.pl/login\[.html?#xxx@xxx.gov.xx](https://controladmin.7m.pl/login[.html?#xxx@xxx.gov.xx), which induces the victim user to type in the username and password. The web page will, in the end, return to the enterprise homepage that the victim user works.

On the hacker side, the back-end server will receive the event log of the failed login attempts from the victim user, and then record the username and password. Hence, the hacker can use the legitimate email account to sign in, such as web page or email server, and can even further send an elaborately crafted phishing email to a person who is the victim’s frequent contact, which is hard to be detected by most security products.

Case C: Phishing from a Legitimated Website: The malicious execution flow shown in Fig. 2(c) is similar to the attack shown in Fig 2(b) in that it also has a link embedded in the mail content for phishing campaign. However, the phishing link [https://armonaoil.com/admin/images/npgtr/news/potcpanel\[.html](https://armonaoil.com/admin/images/npgtr/news/potcpanel[.html) is from a legitimate website rather than from a personally created malicious website, which indicates that the legitimate website has been compromised for the use of darknet market¹.

By further analysis, we found that the enterprise in this case indeed opened the cPanel web hosting server for public access, which was vulnerable to the brute-force attack and remote external control. Furthermore, we examined the

¹The darknet is most often used for illegal activities such as black markets, illegal file sharing, and the exchanging of illegal goods or services.

Email Subject	Payload	Microsoft	Tencent	Kaspersky	FireEye	McAfee	Qihoo-360	HOLMES
SF Express New Order_INV 2019022411	MALWARE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
Confirm Your Invoice for Payment	MALWARE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE
Mail Update Notification. Inbox Full on	LINK	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Purchase Order	MALWARE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE
About: Ownership Confirmation of***	LINK	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Reminder: Your Package Could Not Be Delivered***	LINK	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Re: **TOP URGENT** BL Draft Copy	MALWARE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
REE:URGENT QUOTATION NEEDED ASAP	MALWARE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
Re:QUOTATION TEMPLATE2021	MALWARE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
***disconnected Fix Now!	LINK	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Validate your Password for***	LINK	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Your email***will be closed soon	LINK	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Notifications undelivered emails to your mailbox	LINK	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
DHL BILL OF LADING SHIPPING INVOICE	MALWARE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
Attention***mail upgrade	LINK	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

Fig. 3. Detection Result of HOLMES Compared with Other Detectors in VirusTotal Enterprise Version

recent activities on some popular darknet markets and found that more than 3,000 sites of cPanel accesses were selling in the darknet market (raidforums) since 2020-11-22. Hence, it can be confirmed that the email attack is a phishing campaign caused by the third-party information leakage.

B. A Comparative Study

To evaluate the detection capability of HOLMES, we compare it with some commercial email detectors that are offered by six key security vendors in VirusTotal: Microsoft, Tencent, Kaspersky, FireEye, McAfee and Qihoo-360. We select 15 malicious emails as testing samples. They either contain a phishing link or have a malware infected attachment. All the testing samples were collected from the real-world threat hunting during the whole December month in 2020, and these samples had bypassed the detection of the enterprise anti-spam gateway and successfully detected by HOLMES. More examples can be found in Table I. The comparison is to demonstrate the proportion of highly-concealed malicious emails, which the other commercial tools still cannot to discover. The result shows the evidence and reason why we still need a behavioural anomaly detection tool like HOLMES for anomalous email detection.

The comparison table is given in Fig. 3, where the email subjects representing the 15 malicious emails are listed in the first column and the rest columns are the detection results from the commercial detectors and HOLMES. A FALSE value from a detector on a malicious email indicates that the detector failed to identify the malicious email.

For the detectors of Microsoft, Kaspersky and FireEye, we can see a good performance on detection of those malicious emails that contain malware infected attachments. However, they fail to detect the malicious emails that contain phishing links. Based on the further analysis by our security experts, most of the phishing domains have been registered no more than three months and some of them are even from legitimate known enterprises. Moreover,

all the phishing links include a specific URL to access the particular crafted phishing web page under the domain name that is shortly expired in around three days. Such a short-lived situation significantly increases the difficulty of anomaly detection.

From the comparison table, we can also see that McAfee demonstrates a moderate detection rate on the malicious emails that contain malware infected attachments. Similar to Microsoft, Kaspersky and FireEye, McAfee also cannot detect the malicious emails that contain a newly registered phishing link.

Compared to all above detectors, Tencent and Qihoo-360 have a low detection rate. Among the 15 malicious emails, only two are detected by Tencent and three by Qihoo 360.

We would clarify that, the detection engines used for the comparison are supplied by the VirusTotal Enterprise Service. Since the version of the detectors may not be the same used in their commercial products, we would state that the comparison result cannot completely indicate the detection capability of their latest versions in the commercial products.

V. ENHANCEMENTS IN THE LATEST IMPLEMENTATION

After the first deployment to the enterprise environment, as mentioned in the above section, HOLMES has been upgraded with a few enhancements. In the latest version of HOLMES, we rebuild the code warehouse that makes HOLMES more efficient to discover anomalies in a much smaller rolling time window. The improvement is achieved by moving the data query system from Elastic-Search (ES) server to the real-time Kafka computing platform. The main difference between ES and Kafka is the way the data is processed. ES uses batch processing whereas Kafka uses stream processing, and the stream processing is more timely and efficient. The advantage of using Kafka is that HOLMES can detect anomalies in less than one minute without the risk of server crash, significantly reducing the computing

TABLE I
 SAMPLES OF PHISHING EMAILS DETECTED BY HOLMES ON 2021-10-26.

Phishing Emails Detected by HOLMES on 2021-10-26		
<p>Subject: You have an outstanding payment.</p>	<p>Subject: Please confirm your email account.</p> <p>Please confirm your email account</p> <p>gov.cn <gov.cn Support <no-reply...>> 2021年10月4日 星期一 下午9:37</p> <p>收件人: gov.cn</p>	<p>Subject: Incoming mails has blocked!</p> <p>Incoming mails has blocked!</p> <p>gov.cn <support@dgysme3.gov.uk> 2021年9月30日 星期四 下午8:45</p> <p>收件人: gov.cn</p>
<p>Subject: Closure Of Mailbox Notice !</p> <p>Closure Of Mailbox Notice !</p> <p>Account Server <expo@...> 2021年10月5日 星期二 上午10:02</p> <p>收件人: expo@...</p> <p>Mailbox Update Notice.</p> <p>Hello expo@...</p>	<p>Subject: Security notice - Immediate action required.</p> <p>Security notice - Immediate action required</p> <p>E-Mail Server <noreply@etigr.com> 2021年10月5日 星期二 上午9:17</p> <p>收件人: gov.cn</p>	<p>Subject: Security notice - Immediate action required.</p> <p>Security notice - Immediate action required</p> <p>E-Mail Server <noreply@etigr.com> 2021年10月5日 星期二 上午10:35</p> <p>收件人: gov.cn</p>
<p>Subject: xxx@xxx.gov.cn Protection</p> <p>gov.cn 维护保护</p> <p>邮箱部 <jiangxur@you.com> 2021年10月2日 星期六 上午11:30</p> <p>收件人: jiangxur@you.com</p>	<p>Subject: Email Account Security Alert Request.</p> <p>Email Account Security Alert Request</p> <p>Postmaster <Mail-SecurityTeam@noreply...> 2021年9月30日 星期四 下午10:03</p> <p>收件人: gov.cn</p>	<p>Subject: Notification: xxx@xxx.gov.cn Disk is full.</p> <p>Notification: gov.cn Disk is full</p> <p>gov.cn (C) <sm@milax.cf> 2021年10月5日 星期二 下午5:44</p> <p>收件人: gov.cn</p>
<p>Subject: [WARNING!!!]Your Email Account Is About To Be Terminated.</p> <p>[WARNING!!!]Your Email Account Is About To Be Terminated</p> <p>gov.cn <noreply@3-shippng.com> 2021年10月5日 星期二 上午4:21</p> <p>收件人: gov.cn</p> <p>ACCOUNT TERMINATION IN PROGRESS</p>	<p>Subject: xxx@xxx.gov.cn Password Update.</p> <p>gov.cn Password Update</p> <p>Mailbox Support <gov.cn> 2021年10月5日 星期二 上午11:33</p> <p>收件人: gov.cn</p> <p>Your password has expired</p>	<p>Subject: One time verification.</p> <p>One time verification</p> <p>Mail Service <info@sender.com> 2021年10月9日 星期六 下午12:30</p> <p>收件人: gov.cn</p>

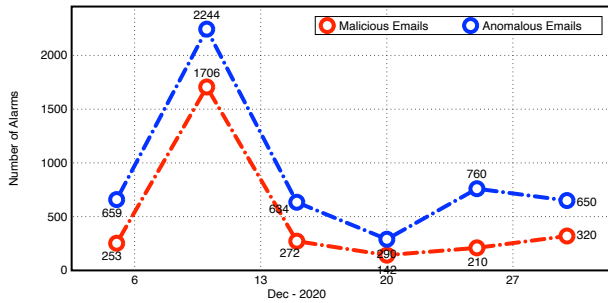


Fig. 4. Detection Performance (December 2020)

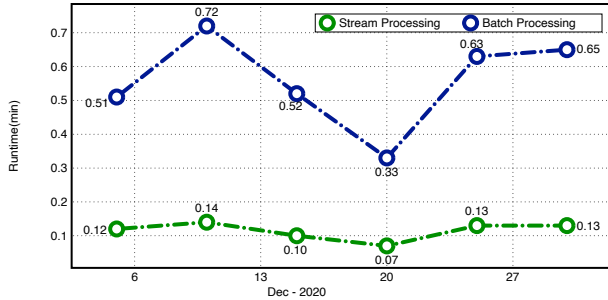


Fig. 5. Stream Processing & Batch Processing

consumption. Fig. 4 and Fig. 5 respectively shows the detection performance of HOLMES in December 2020 and the run-time performance improvement after switching batch processing to stream processing. Furthermore, use of Kafka can also help our security analysts to better schedule time for threat identification and improve the efficiency of threat responses.

VI. CONCLUSION

In this paper, we introduce HOLMES, a lightweight semantic based anomalous email detector, which can effectively discover malicious emails in the real-world cyber threat hunting. HOLMES also demonstrates a viable solution that successfully transfers AI technology to the cyber security field and makes an excellent trade-off between the cost of algorithmic consumption and the detection performance.

We measure the performance of HOLMES, and compare its detection capability with several well-known commercial detectors offered by the security companies in VirusTotal. Our evaluation result shows that, on the use of cyber threat hunting, HOLMES significantly outperforms those commercial products in a range of malicious attack scenarios, which demonstrates its practical values in the commercial competition. Our current evaluation is based on one-month data. As future work, we will extend our investigation on a large scale dataset that is collected over a long test period and covers emails of different languages.

REFERENCES

[1] I. D. Foster, J. Larson, M. Masich, A. C. Snoeren, S. Savage, and K. Levchenko, "Security by any other name: On the effectiveness

of provider based email security," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 450–464, 2015.

[2] VirusTotal. "https://www.virustotal.com/gui/home/search". (accessed: 11.25.2020).

[3] MITRE. "https://attack.mitre.org/matrices/enterprise/". (accessed: 11.25.2020).

[4] G. Ho, A. Cidon, L. Gavish, M. Schweighauser, V. Paxson, S. Savage, G. M. Voelker, and D. Wagner, "Detecting and characterizing lateral phishing at scale," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 1273–1290, 2019.

[5] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," 2010.

[6] Barracuda. "https://www.barracuda.com/glossary/email-spoofing". (accessed: 11.25.2020).

[7] M. Wong and W. Schlitt, "Sender policy framework (spf) for authorizing use of domains in e-mail, version 1," tech. rep., RFC 4408, April, 2006.

[8] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas, "Domainkeys identified mail (dkim) signatures," tech. rep., RFC 4871, May, 2007.

[9] M. Kucherawy and E. Zwicky, "Domain-based message authentication, reporting, and conformance (dmarc)," ser. *RFC7489*, 2015.

[10] J. Chen, V. Paxson, and J. Jiang, "Composition kills: A case study of email sender authentication," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.

[11] P. Wu, N. Moustafa, S. Yang, and H. Guo, "Densely connected residual network for attack recognition," *19th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom)*, 2020.

[12] P. Wu, H. Guo, and N. Moustafa, "Pelican: A deep residual network for network intrusion detection," in *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 55–62, IEEE, 2020.

[13] P. Wu and H. Guo, "Lunet: A deep neural network for network intrusion detection," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 617–624, IEEE, 2019.

[14] W. U. Hassan, A. Bates, and D. Marino, "Tactical provenance analysis for endpoint detection and response systems," in *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1172–1189, IEEE, 2020.

[15] I. U. Haq, I. Gondal, P. Vamplew, and S. Brown, "Categorical features transformation with compact one-hot encoder for fraud detection in distributed environment," in *Australasian Conference on Data Mining*, pp. 69–80, Springer, 2018.

[16] P. Cerda, G. Varoquaux, and B. Kégl, "Similarity encoding for learning with dirty categorical variables," *Machine Learning*, vol. 107, no. 8, pp. 1477–1494, 2018.

[17] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.

[18] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, pp. 1188–1196, 2014.

[19] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 1649–1652, 2009.

[20] Swaks. "https://github.com/jetmore/swaks". (accessed: 11.25.2020).

[21] CobaltStrike. "https://www.cobaltstrike.com/". (accessed: 11.25.2020).

[22] CVE-2017-11882. "https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2017-11882". (accessed: 11.25.2020).