Zhao *et al. Cybersecurity* (2023) 6:36
https://doi.org/10.1186/s42400-023-00167-8

Cybersecurity

RESEARCH

Open Access

Detecting compromised email accounts via login behavior characterization

Jianjun Zhao[1,2], Can Yang[1,2], Di Wu[3], Yaqin Cao[1,2], Yuling Liu[1,2], Xiang Cui[4] and Qixu Liu[1,2*]

Abstract

The illegal use of compromised email accounts by adversaries can have severe consequences for enterprises and society. Detecting compromised email accounts is more challenging than in the social network field, where email accounts have only a few interaction events (sending and receiving). To address the issue of insufficient features, we propose a novel approach to detecting compromised accounts by combining time zone differences and alternate logins to identify abnormal behavior. Based on this approach, we propose a compromised email account detection framework that relies on widely available and less sensitive login logs and does not require labels. Our framework characterizes login behaviors to identify logins that do not belong to the account owner and outputs a list of account-subnet pairs ranked by their likelihood of having abnormal login relationships. This approach reduces the number of account-subnet pairs that need to be investigated and provides a reference for investigation priority. Our evaluation demonstrates that our method can detect most email accounts that have been accessed by disclosed malicious IP addresses and outperforms similar research. Additionally, our framework has the capability to uncover undisclosed malicious IP addresses.

Keywords Compromised account detection, Mixture model, Login log analysis, Attribution and forensic

Introduction

Email is one of the most widely used essential tools in modern enterprise settings. Email correspondence can reveal an enterprise's personnel structure, and email content can reveal employees' work content. Both of them are usually sensitive information in the interests of adversaries. Phishing emails and data breaches are the most significant causes of email account compromise. In the first half of 2022, cyberattacks against email increased by 48%, and nearly 70% of those attacks included a credential phishing link (Corp 2022). Additionally, continuous data leakage provides adversaries with a large number of email accounts and passwords (UpGuard 2022), which can be automatically verified through scripts, enabling adversaries to acquire many compromised email accounts. Once obtained, adversaries can easily steal an enterprise's commercial secrets and move laterally based on staff relationships. The longer these compromised accounts exist, the more harmful they are to the enterprise. Therefore, it is essential to detect such compromised accounts promptly. For blue teams, it is crucial to attribute attacks and locate their source and organization.

Most research on detecting compromised accounts focuses on social network settings, where there are various interaction events such as logging in, posting, and liking. These events provide numerous features, enabling researchers to detect compromised accounts using various methods. However, for email accounts, there are only two interaction events: sending and receiving. These features can be acquired from login behaviors (SMTP for

*Correspondence:
Qixu Liu
liuqixu@iie.ac.cn
[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3] China Cybersecurity Review Technology and Certification Center, Beijing 100013, China
[4] Zhongguancun Laboratory, Beijing 100089, China

sending, IMAP/POP3 for receiving, and Webmail for both). Therefore, interaction features that can be used to detect compromised email accounts are all included in the login logs. On the other hand, the login logs contain all the attack actions. EvilCohort (Stringhini et al. 2015) uses login logs to detect malicious accounts, but its main detection objects are fake accounts in social networks rather than compromised email accounts of employees in enterprise settings.

To address this gap, we propose a Compromised Email Accounts Detection framework (CEAD), which can detect abnormal login behaviors of email accounts and output a list of account-subnet pairs for security teams. Compared to spam/phishing email detection, which marks abnormal results with specific labels, detecting compromised email accounts is more challenging due to the lack of personal confirmations from the account owners or reliable credible Indicators of Compromise (IoCs). Determining whether an email account is compromised based on a single login is impossible. To overcome this challenge, CEAD narrows the scope requiring manual investigation as much as possible and provides suspicious lists and ranks instead of a specific diagnosis.

Our approach characterizes the login behavior of email accounts from both temporal and spatial perspectives. To characterize temporal behavior, we refer to the idea of building a Gaussian mixture model (GMM) (Reynolds 2009). We treat login behavior as a mixture model composed of the owner's and potential attacker's models. If the owner and attacker are in different time zones, their distribution in the mixture model will also differ, allowing us to use this feature to detect abnormal behavior. After fitting the login behaviors, if the weight of the owner's model is small, we consider it more likely that there is an attacker. However, it is challenging to find data that can represent the owner in a large number of login events. Therefore, we propose the concept of subnet reputation. We first extend the basic unit of the login address from IP to subnet and then calculate subnet reputation based on three activity-related features. We use subnets with higher reputations to construct the owner's model. If the weight of the owner's model decreases after a new subnet joins the mixture model, it suggests that the newly joined subnet behaves differently from the owner's, indicating that it may be suspicious.

When the owner and attacker log in to the same account together for a period of time, the login location will frequently change, and the accumulated distance between login locations will be very large. Therefore, we can use this feature to characterize abnormal spatial behaviors. However, if the owner also frequently changes the IP address (e.g., using a proxy), it is impossible to distinguish between the two situations. Hence, we use entropy to measure whether changes in login locations are common, i.e., a compromised email account may have a login series with drastic but not common location changes.

We evaluated the performance of our framework on nine datasets, consisting of over 109 million login events. Our approach effectively reduces the number of account-subnet pairs that security teams need to investigate. We rank account-subnet pairs based on their likelihood of having abnormal login relationships, and performs better than DAS (Ho et al. 2017). An additional advantage of our framework is that it relies only on login logs, significantly reducing the exposure of other sensitive information, such as relationships and email content. This feature is particularly relevant in enterprise settings where security services are outsourced. Moreover, our framework is self-sufficient, not requiring any labeled data or prior knowledge, and can work independently without other security detection measures. In addition, our framework provides flexibility as some of its components or mechanisms can be replaced or modified. For instance, if analysts have a more reliable reputation calculation mechanism, they can directly replace our reputation calculation method. Similarly, if analysts want to experiment with a new ranking algorithm, they can modify the corresponding formula with ease.

In summary, this paper makes the following contributions:

- We propose a new method of detecting abnormal login behavior, noting that the two main characteristics, time zone differences and alternate logins, are difficult for an attacker to circumvent simultaneously.
- We present CEAD, a compromised email accounts detection framework that uses only login logs and does not rely on labeled data. CEAD detects both temporal and spatial behavior anomalies, significantly reduces the burden of analysts when investigating.
- We evaluated CEAD on email login logs from nine organizations. CEAD sorted account-subnet pairs by likelihood of having abnormal login relationships, successfully detected most compromised emails and undisclosed malicious IP addresses, and performed better than DAS.

Background

The illicit use of compromised email accounts by adversaries has grave consequences, including the propagation of false information (CNN 2021) and political manipulation (Wikipedia 2022b). Advanced persistent threats (APTs) (Wikipedia 2022a) are often attributed to

nation-state or state-sponsored groups, with clear and specific objectives. These adversaries may target intelligence after seizing control of email accounts. In the event that the victim is not the primary target or holds no significant value, attackers may use the compromised email account to move laterally and expand their target.

Prior studies have predominantly concentrated on detecting spam and phishing emails. Among them, methods for detecting lateral movement phishing emails can be used to detect compromised email accounts. However, these techniques operate under the assumption that attackers engage in sending behaviors. In instances where adversaries aim to steal emails, such methods cannot detect compromised email accounts. Our study pertains to detecting compromised user credentials (CUCs), and many related studies model and compare the usage behavior of compromised accounts, especially in the domain of social networks. Our work draws upon similar ideas to these studies.

Related works

Current research on email security has primarily centered on detecting and filtering spam/phishing emails (Ho et al. 2017, 2019; Hu et al. 2016). Spam/phishing emails detection and compromised email accounts detection are different scenarios in the same field. The former focuses on identifying email content and real-time defense capability, while the latter concerns detecting attack outcomes and providing support for attribution and forensics. Our work falls into the latter category.

Ho et al. (2017) presented a novel approach for detecting compromised email accounts resulting from phishing attacks. Their methodology utilizes datasets including email samples, HTTP logs, and login logs to assess the likelihood of compromise based on both the sender's and domain's reputation. Through the use of HTTP logs, the authors track user behavior upon accessing phishing links and determine the status of the account based on whether the user entered their password.

Hu et al. (2016) proposed a method for identifying compromised accounts through analysis of abnormal email correspondence relationships as indicated by sending and receiving logs. However, this approach may fail to detect certain compromised accounts if the attacker's sole objective is to steal emails without engaging in lateral movement, as the correspondence relationship remains unchanged.

In essence, techniques designed to detect lateral movement phishing attacks can also be applied to identifying compromised email accounts. Ho et al. (2019) employ three key features, namely the similarity of email recipients, the reputation of senders, and the reputation of URLs, to detect lateral movement attacks. The authors determine the success of an attack by examining the causality between malicious email senders.

In addition to email accounts, other online web services are also at risk of account compromise. Attackers may seek to illegally exploit social networks for significant gains, leading to a rise in various attacks against social network accounts. Currently, there are numerous approaches available for detecting compromised accounts on social networks (Ruan et al. 2015; Viswanath et al. 2014; Egele et al. 2015; Stringhini et al. 2015; Karimi et al. 2018; Egele et al. 2013; Pv and Bhanu 2020; Velayudhan and Somasundaram 2019).

Ruan et al. (2015), Viswanath et al. (2014), and Egele et al. (2015) have characterized the behavior of social network users to identify actions that deviate from the norm. Their approaches leverage temporal and spatial features, such as user operating time and application clients, to detect abnormal user behavior. Stringhini et al. (2015) proposed a more generalized method for detecting compromised accounts across various online network services. Their approach involves constructing relationships between operating events and IP addresses using a bipartite graph, followed by clustering based on the one-mode projection of the graph to identify communities of compromised accounts.

Challenges

Advanced persistent threat (APT) groups demonstrate a higher degree of sophistication and caution in their attacks compared to adversaries with profit-driven motives. APT groups are more likely to leverage dynamic IP addresses and minimize unnecessary attack actions. Consequently, detecting related compromised email accounts presents several challenges.

Challenge1: concealed access actions

In certain instances, adversaries may opt to steal their victims' emails instead of leveraging their accounts for lateral movement. This strategy is particularly favored when targeting high-value victims, as it allows the adversary to maintain long-term access to the compromised email accounts without being detected. For email-theft attack campaigns, the attack behavior will only manifest in the login logs, which typically contain limited information. Consequently, detecting this type of attack can be highly challenging.

In our detection results, none of the malicious IP addresses have the behavior of sending emails, such as logging in via SMTP. This implies that features such as email correspondence, email headers, email bodies, and attachments can not be used to detect this type of the

attack. Consequently, the only available data to detect this type of attack is the login logs. Therefore, some previously employed methods for detecting spam/phishing emails are not applicable in this scenario.

To overcome this problem, we focus on login events. We utilize four common features (protocol, datetime, IP address, and email account) that are present in both information-stealing and lateral movement attacks to characterize the login behavior.

Challenge2: mixed benign and adversarial location-changed logins

It is common for employees to log in to their business email accounts from non-frequently used places, particularly in large international companies and organizations. For instance, an employee traveling for business purposes may use the local network or unintentionally or intentionally use a proxy, such as logging in to webmail after visiting a website with a proxy-configured browser or setting up a proxy directly for the client due to network requirements. On the other hand, when APT groups log in to their victims' accounts to steal emails or send lateral phishing emails, they may use IoT bot-nets, compromised hosts, and easily accessible and disposable cloud servers as their attack infrastructure or springboard. As depicted in Fig. 1, most accounts have a percentage of location-changed logins, which may be contributed by both the owner and the attacker.

To distinguish mixed behaviors and identify anomalies, previous research has focused on characterizing the operating habits of different individuals, including their sessions and devices (Ruan et al. 2015). Similarly, in our scenario, we can detect anomalies by comparing the login behavior of previously unseen IP addresses with that of benign IP addresses.

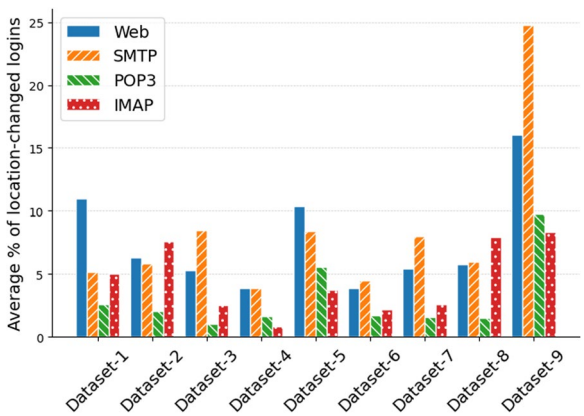

Fig. 1 The average percentage of location-changed logins for each login protocol in our datasets

Challenge3: limited indicators of compromise

Many existing methods rely on labels or prior knowledge when determining whether an IP address is malicious or not. For instance, Stringhini et al. (2015) employ a labeled dataset provided by an email service provider, while other researchers may use customized rules for manual labeling (Ho et al. 2019). However, manual labeling requires a sufficient amount of features in the data to allow researchers to distinguish between malicious and benign behavior. Data from a single source, such as login logs, often has limited information and is challenging to label manually.

IoCs play a crucial role in attributing attack campaigns to APT groups. These IoCs are derived from the forensic practices of numerous security researchers and provide reliable information. However, the number of IoCs available is far from sufficient. Threat intelligence can only disclose a portion of the IP addresses used by APT groups, and adversaries may use different IP addresses when attacking different targets. This limited availability of IoCs can leave security analysts with a dearth of references when conducting investigations.

To address this problem, DAS (Ho et al. 2017) generates a list of events ordered by their level of suspicion, instead of providing a conclusive decision. Similar to the this approach, we provide analysts with a priority when manually investigating.

Methodology

In this section, we introduce our two key ideas for detecting compromised email accounts. First, we leverage the distributions of login times to construct a mixture model and utilize the weight of the benign model to determine whether the account has been compromised. Second, we identify suspicious login sequences in which there is a drastically but unusual change in the login location. These two characteristics capture abnormal email usage from different perspectives, and we demonstrate that it is challenging for an attacker to circumvent both simultaneously.

Time zone difference

Attacks against transnational targets may result in login behaviors spanning two or more circadian cycles in different time zones. Figure 2 presents a typical example, showing the distribution of both the owner and the attacker logging into the same email account from 0 to 24 o'clock. We estimated their distributions via kernel density estimation (KDE) (Wikipedia 2022c) based on login logs, and for privacy reasons, we have hidden the specific time and shifted the distributions. The M-shape observed in the figure reflects the typical working time of approximately 8 h per day, with two peaks and a trough

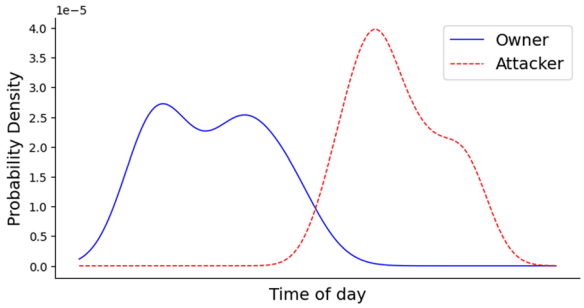

Fig. 2 In a real case in our datasets, the account was logged in by both the owner and the attacker, who were in different time zones

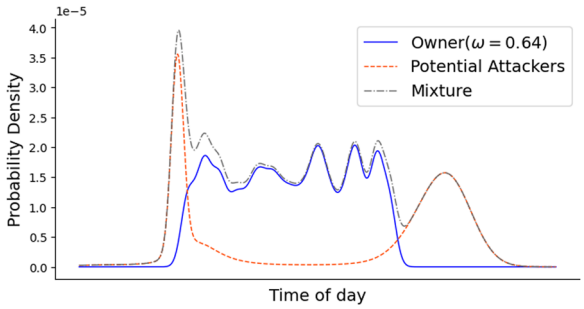

Fig. 3 In a real case in our datasets, the weight of the M_{Owner} is only 0.64, which means that there is a different login behavior, represented in the figure as *Potential Attackers*

corresponding to morning work, afternoon work, and lunch break.

Mixture model: We consider the distribution reflected by all the data in the login logs as a mixture model composed of the distributions of the email account owner and the potential attackers. Suppose we denote by M_{Owner} the probability distribution of owner's logins and by $M_{PotentialAttackers}$ the probability distribution of potential attackers' logins. In that case, we can express the mixture model as

$$M_{Mixture} = \omega M_{Owner} + (1-\omega) M_{PotentialAttackers} \quad (1)$$

where ω denotes the weight, i.e., the probability that a particular login contributed by the owner. From a holistic perspective, ω represents the similarity between the owner's model and the mixture model. Therefore, the smaller the weight of M_{Owner}, the more likely there is an attacker, and vice versa. Note that the distribution of M_{Owner} is not necessarily M-shaped as described above and may be any shape depending on one's working habits and the login protocol used (Webmail, POP3, SMTP, IMAP). Figure 3 shows an example of a mixture model. We hide the horizontal coordinate scale in this figure for the same reason as in Fig. 2.

Table 1 Four possible cases of the mixture model and the corresponding determination results

	Low credibility reference model	High credibility reference model
Large ω	3.suspicious	1.normal
Small ω	4.suspicious	2.suspicious

Two-factor determination combining credibility and weight: Without prior knowledge or an IP address whitelist, it is challenging to select the data that can represent the owner from all the logins to build the M_{Owner}. Thus, we select some IP addresses with more logins and use their logins to build a reference model and evaluate the model's credibility based on the reputation of these IP addresses. Now the mixture model can be expressed as

$$M_{Mixture} = \omega M_{Reference} + (1-\omega) M_{Others} \quad (2)$$

There are two reasons for using two-factor determination: first, there may not be enough logins from high-reputation IP addresses to build M_{Owner}; second, it is not reasonable to directly regard the model built based on the IP addresses with more logins as M_{Owner} because the attacker may also contribute many logins (e.g., using a client or script that automatically steals emails).

Table 1 presents four possible cases of the mixture model. In the first case, where the reference model has high credibility and a large weight ω, the existence of an attacker is considered less likely. In the second case, where the reference model also has high credibility but a small weight ω, the mixture model appears suspicious as it no longer resembles the high credibility reference model after adding the login data of the remaining IP addresses. However, in the second case, it can only reflect a different login behavior from the reference model but not gives a definitive decision because both the attacker and the owner may contribute the remaining data. For instance, the reference model is based on login data from IP addresses in the employee's workplace during the day, while the employee uses different IP addresses to check emails after work. Therefore, we will use other methods to filter further. We consider the third and fourth cases suspicious because the logins used to build the reference model (most of the logins) have low reputations.

Alternate logins

We can obtain a series if we sort all the logins by time and take the spatial attributes of logged-in IP addresses as elements. Figure 4 provides a simple example. If only the owner logs in to their email account over a certain period, we will observe a relatively stable series. In

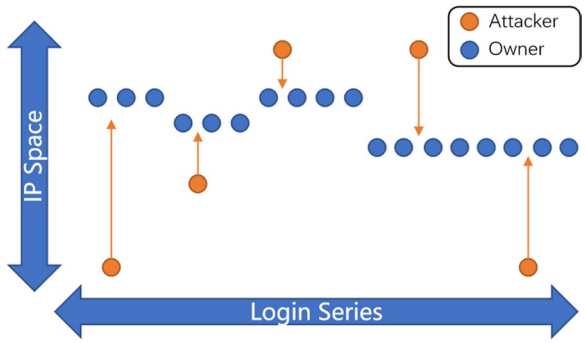

Fig. 4 The alternating logins of the attacker and owner generate a login series with extreme volatility

contrast, if both the owner and the attacker log in simultaneously during that period, we will observe a volatile series.

Distance series: We can use different IP address attributes to represent the IP space, such as the geographic location of an IP address or the Autonomous System (AS) to which an IP address belongs. However, if the IP addresses used by the attacker and the owner belong to the same AS or are geographically close, the extreme variation that should exist is lost. We believe that the former is more likely than the latter. For instance, both the attacker and the owner might use the cloud service of the same ISP, but it is relatively rare that the attacker uses an IP address close to the owner. Therefore, we opt to use the geographic location of the IP address to represent spatial features. The latitude/longitude of an IP address constitutes a two-dimensional feature. By calculating the distance between geographic locations, we can obtain a one-dimensional series convenient for measurement. There are several methods to convert the latitude/longitude series into a distance series, such as calculating differential distances or cumulative travel distances.

Anomaly evaluation: Various indicators can be employed to assess the volatility of a series, such as standard deviation and coefficient of variation (CV). Nonetheless, these indicators can only reflect the degree of extreme volatility in a series, and not its frequency. To evaluate the commonality of volatility, entropy can be utilized to determine the complexity of a series. A series with high complexity corresponds to a larger entropy value and vice versa. In the scenarios under study, attackers are less likely to carry out attacks frequently to minimize exposure risk. Hence, if the volatility in a series is frequent, we tend to assume that the owner is responsible for it (e.g., using proxies). Combining standard deviation and entropy, we can identify a series with extreme, yet uncommon, volatility. That is, when the login location is typically stable, but undergoes frequent changes during a particular time, we can infer that the account may have been compromised.

Complementarity

When the time zones of the attacker and the owner are close, it may be difficult to distinguish them based on temporal characteristics alone. In such scenarios, it is highly likely that the owner and attacker have alternate logins. Therefore, spatial characteristics can still be used to detect compromised accounts. Conversely, if the attacker and the owner do not have alternate logins, they may be in different time zones. In this case, temporal characteristics can be used to detect the compromise.

As a result, both aspects mentioned above must be considered when detecting compromised accounts. We have designed a framework to characterize both temporal and spatial behaviors, which we will describe in detail in the following section.

Framework design

Based on the above understanding, we propose CEAD, a compromised email accounts detection framework. As shown in Fig. 5, CEAD contains two main modules for

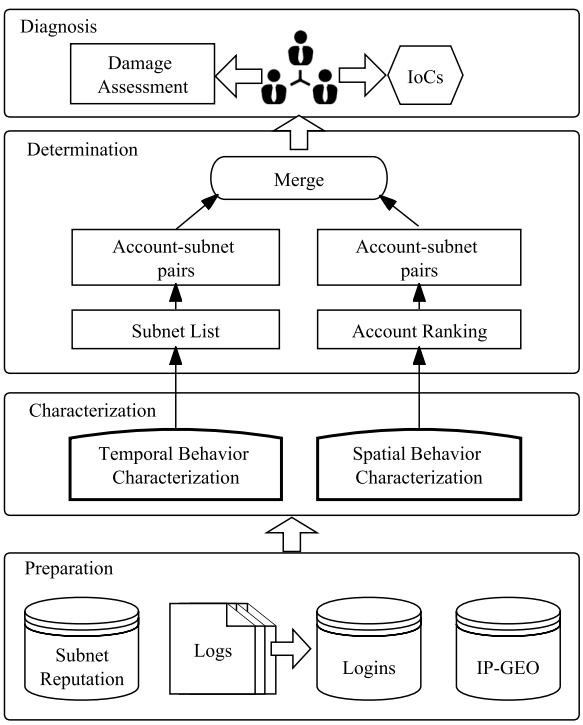

Fig. 5 The architecture of CEAD

characterizing logins' temporal and spatial behaviors, two mechanisms for determining anomaly of characterization results. First, we introduce the method of building and fitting a mixture model and the method of screening suspicious subnets. Second, We introduce the method of measuring the spatial variation in login behavior. Lastly, we illustrate the use of CEAD.

Temporal behavior characterization

As previously mentioned, we expect to build a mixture model to describe the temporal distribution of login behaviors and to evaluate the similarity between the reference model and the mixture model based on the weight. By incorporating the reliability of the reference model, we can estimate the likelihood of an account being compromised.

Application of GMM and EM

The Gaussian mixture model (GMM) (Reynolds 2009) can be applied to our work with minor modifications, as it is capable of fitting almost any distribution. Additionally, the expectation maximization (EM) algorithm (Xu and Jordan 1996) is applicable in our scenario, as it is often used to estimate the parameters of GMMs.

GMM utilizes K Gaussian distributions with distinct parameters to fit the data, where each Gaussian distribution has its own ω, μ, and σ. Here, ω represents the weight of the Gaussian distribution in the mixture model, and μ and σ denote the mean and standard deviation, respectively. In our scenario, we expect a mixture model like Eq. 2, so we use the $M_{Reference}$ to replace the first Gaussian distribution and fit the M_{Others} with the remaining $K-1$ Gaussian distributions. Thus the mixture model can be expressed as

$$M_{Mixture} = \omega_0 M_{Reference} + \sum_{k=1}^{K-1} \omega_k N(\mu_k, \sigma_k) \tag{3}$$

where $\sum_{k=0}^{K-1} \omega_k = 1$.

Building the reference model: We use the logins from the most active IP addresses as the primary data and employ kernel density estimation (KDE) (Wikipedia 2022c) to estimate the reference model. Initially, we count the number of logins from each subnet for each account during a specified time period, such as one subnet's lifetime. Next, we choose the most active subnets, based on the number of logins, and use their login events as the basic data. We convert the login time to seconds of a day, ignoring the date, to create a list L, where each element falls within the range of 0–86,400. KDE can estimate the probability distribution based on discrete data, and hence, we can estimate the reference model using L as the input for KDE.

We chose to use subnets as the base unit because using individual IP addresses could result in too many small data fragments due to the use of DHCP, even if the IP addresses belonged to the same person, making it challenging to construct a model. In selecting the subnet size, we cannot directly determine it based on autonomous systems (AS) since some subnets may be too large, with over a million IP addresses. Nur and Tozal (2018) showed that the /24 is the most commonly used prefix, and the attacker and the owner are less likely to use the same 24/ subnet. Hence, we considered it reasonable to choose the /24 prefix as the subnet size.

Estimating the parameters: The advantage of utilizing Eq. 3 is that it does not alter the EM estimation process for the μs, σs, and ωs in GMM, even though the reference model does not include μ and σ. In our scenario, we are particularly interested in the results of ω_0 rather than the μs, σs, and ωs for the other Gaussian distributions.

Initial parameters: The EM algorithm requires initial parameters before starting the iteration, including K, $\omega_0 ... \omega_k$, $\mu_0 ... \mu_k$, $\sigma_0 ... \sigma_k$. Different initial parameters may result in distinct final fitting results. However, since we are primarily concerned with the resulting weights rather than the model's fitness, we will not excessively optimize the model fit by adjusting these hyperparameters. Therefore, we provide a guideline for setting these initial parameters.

First, we consider $K = 11$ as sufficient, as we use $K-1$ Gaussian distributions to describe the M_{Others}, which only contains a small amount of data and does not require a large K. Second, ω_0 should be set to a large value, such as 0.99, which assumes that all logins initially belong to the reference model, and may gradually decrease as the EM algorithm iterates if some of the logins do not belong to the reference model. Third, for $K-1$ Gaussian distributions, the weights are equally divided $(1-\omega_0)$, and μs are uniformly spaced on the range of 0–86,400. Additionally, all Gaussian distributions have the same σ, and the value should not be too small, as this may lead to a low probability of a certain login on a certain Gaussian distribution and cause the EM program to exit prematurely.

Subnet reputation

Since the reference model is built based on the logins of the most active subnets, it is important to comprehensively consider the credibility of these subnets. Ho et al. (2017) proposed an evaluation method for sender reputation that uses two features about logins from a new city, which could be adapted for our scenario with appropriate modifications. However, this method faces several

challenges in complex scenarios. First, the results may not be accurate since all previous data must be assumed to be benign at the beginning of the analysis. Second, it may not yield satisfactory results in certain special scenarios. For instance, if an employee uses a proxy that is only used by themselves, the number of users and logins may be relatively small, resulting in a lower reputation for an otherwise benign behavior.

We propose a method for evaluating subnet reputation based on three features: the average of the cumulative ratio of logged-in days ($Feature_A$), the average of the cumulative ratio of logged-in times ($Feature_B$), and the coefficient of the number of login protocols($Feature_C$).

Formally, Given a set $\{E_i : 1 \leq i \leq k, k \in \mathbb{Z}^+\}$, where E_i represents the email account logged in by *subnet'*, the $Feature_A$ can be calculated by

$$Feature_A = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{d_{subnet\prime}^{E_i}}{\max\left(D_{subnets}^{E_i}\right)}\right) \tag{4}$$

where $d_{subnet\prime}^{E_i}$ denotes the number of days *subnet'* has logged in to E_i. $\max(D_{subnets}^{E_i})$ denotes the maximum number of the logged-in days of all subnets that have logged in to E_i. Similarly, the $Feature_B$ can be calculated by

$$Feature_B = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{t_{subnet\prime}^{E_i}}{\max\left(T_{subnets}^{E_i}\right)}\right) \tag{5}$$

When a subnet uses a variety of login methods, we consider it to be more trustworthy because for legitimate users, the login protocols are more random. Thus, $Feature_C$ can be calculated by

$$Feature_C = 0.1 \times 2^{the\_number\_of\_login\_protocols-1} \tag{6}$$

We incorporate these three features into our method based on the following rationale: if a subnet logs in many accounts, it will be considered to have a high reputation only if the subnet has more active days and more logins in each account. Conversely, if a subnet logs in to many accounts but is inactive in all of them, it will receive a lower reputation. This is particularly relevant in scenarios where batch attack campaigns are targeting multiple accounts. Additionally, when a subnet uses more protocols, it is deemed more credible. This is because an attacker may not use too many login methods at the same time when trying to steal emails or move laterally, due to purposeful considerations. Furthermore, by using $Feature_C$, we can also filter out some DHCP subnets, as these users have a higher randomness of login protocols.

Based on these three features, we can construct a formula to calculate reputation, as demonstrated in the configuration example in the evaluation section. Other calculation methods can be attempted, such as assigning different weights to each feature and summing them up. Considering the information represented by the features, the reputation result should be positively correlated with all these three features.

Lifetime of subnet

To tackle the issue of unbalanced login data between the owner and attacker, where the attacker's data may be mistaken as noise, we introduce the concept of subnet lifetime. Specifically, we identify a subnet's earliest and latest occurrence time in login logs and analyze the data within this period separately. This allows us to examine whether the behavior of the newly-appeared-subnet during the period of interest differs from the account's primary login behavior.

There are three possible cases that we consider. The first case is when a newly-appeared-subnet has the highest number of logins throughout its lifetime, which leads us to speculate that the user may have changed their primary login method. The second case is when a newly-appeared-subnet does not have the highest number of logins. In this case, we use the login data of subnets with more logins than the newly-appeared-subnet to build a reference model. The third case is when the lifetime of the newly-appeared-subnet is less than one day or the number of logins is too small to build a model. For the second case, we use two thresholds to determine whether the subnet is malicious. For the first and third cases, we assign corresponding labels.

Anomaly determination

After fitting, we obtain a series of ω_0s and reputations, as illustrated in Fig. 6. It should be noted that a single subnet may have multiple points within a subfigure, as it may log in to more than one email account. To classify the results, we use two thresholds, ω' and rep', to divide the data into four areas, corresponding to Table 1. In the results of subnet reputation, high-reputation subnets are indeed trustworthy, but low-reputation subnets may not necessarily be suspicious. This may be due to inaccurate subnet division, resulting in smaller feature values for each subnet. Therefore, we designate the upper-right corner of a subfigure as the trusted area and the lower-right corner as the suspicious area.

Vertical and horizontal comparisons: We can obtain a list of potentially malicious subnets from the suspicious area. However, this list can still be a significant burden for analysts to investigate, so we further filter the

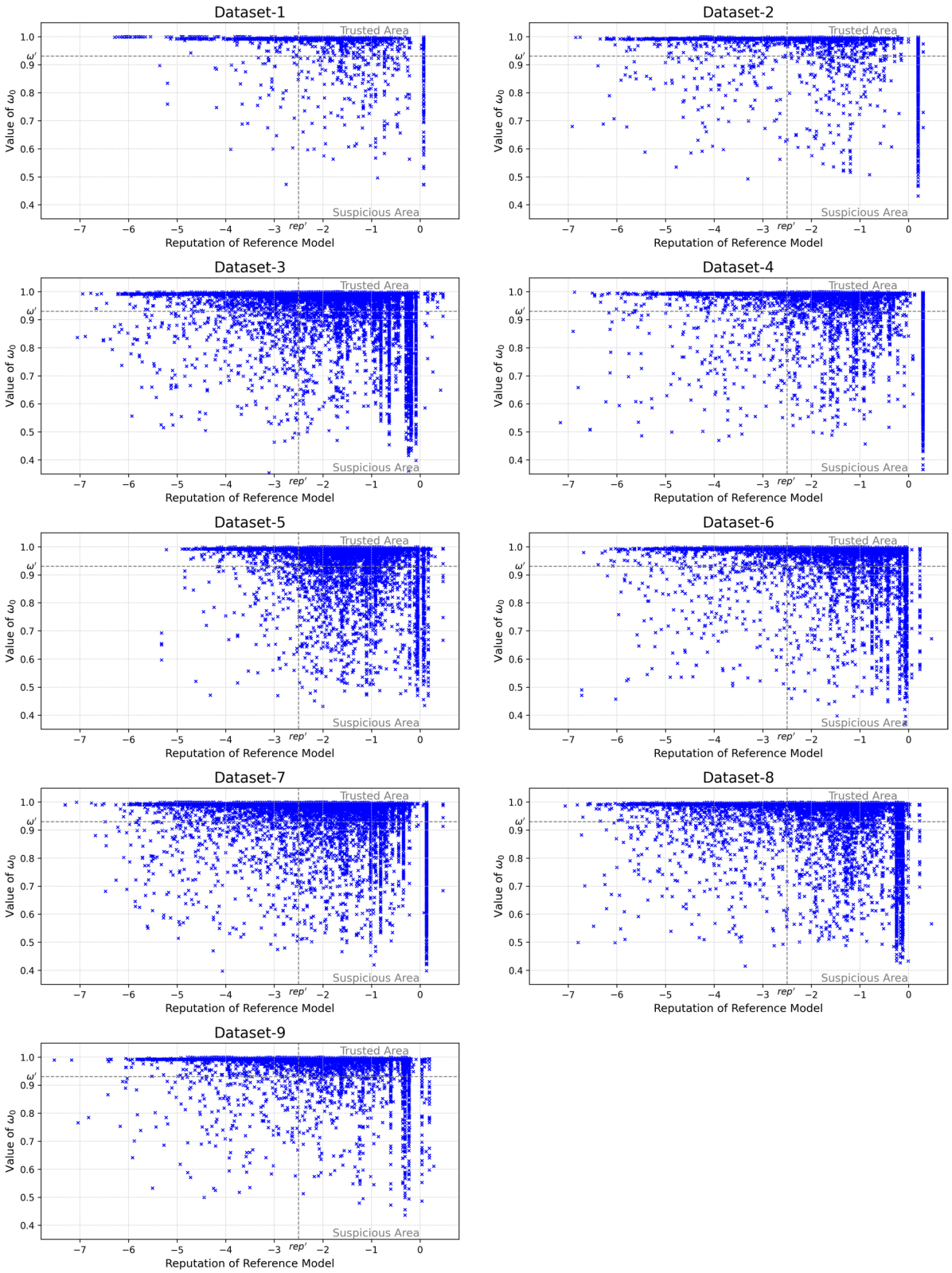

Fig. 6 Temporal characterization result of 9 datasets

suspicious list. First, if a subnet in the suspicious list has other results in the trusted area, we exclude it. In such cases, where a subnet appears abnormal in one account but normal in others, the anomaly may be caused by the user's peculiar usage habit. Second, if a subnet in the suspicious list has a similar distribution to any other trusted subnet that has logged in to the same account, we exclude it. Trusted subnets include subnets with a reputation above the threshold and subnets in the trusted area. In such cases, the anomaly may be caused by the fact that the owner has multiple benign distributions, but the newly-appeared-subnet has been compared to only one of them. To compare whether the distributions are similar, we use Jensen–Shannon divergence (Wikipedia: Jensen–Shannon divergence 2022d).

Spatial behavior characterization

To identify login behaviors that exhibit drastic yet not common changes in IP addresses, we utilize the cumulative travel distance as a metric to quantify the magnitude of the changes and the two-dimensional sample entropy ($SampEn_{2D}$) to determine their level of regularity. Specifically, we focus on login behaviors with larger cumulative travel distances and smaller values of $SampEn_{2D}$, as we suspect these behaviors to be indicative of email account compromise.

Cumulative travel distance

We require a series consisting of appropriate IP address attributes that adequately reflect the spatial difference between the owner and the attacker. Hao et al. (2009) treat IP addresses as a set of numbers from 0 to $2^{32}-1$ and uses the difference as distance, which is inaccurate in some networks. Therefore, we use IP address's geolocation instead of other spatial attributes (such as AS), mainly because it is less likely that the attacker's and the owner's IP addresses are in a nearby geographic location compared to being in the same AS. In addition, the wide use of IP location services makes IP geolocation information easily available, and many websites provide such APIs (ipgeolocation 2022; MaxMind 2022).

We calculate the geographic distances between every two adjacent logged-in IP addresses over a certain period of time (e.g., one hour). By accumulating these distances, we obtain an indicator that measures the spatial variation of the logins, which we refer to as cumulative travel distance. A larger value of this indicator indicates a more drastic change in the login location during the specified time period, and increases the likelihood of an attacker.

Prevalence of spatial variation

Evaluating anomalies based solely on cumulative travel distance is not ideal, as the owner may use a frequently changing network, such as using a proxy pool or using multiple clients simultaneously. Hence, it is essential to assess whether such variations are common in other time periods. Various tools, such as approximate entropy and sample entropy, can be utilized to evaluate the complexity of a one-dimensional series. In our scenario, we need to investigate the complexity of both hourly and daily variations.

$SampEn_{2D}$ (Silva et al. 2016) is well suited to our scenario, which is usually used to assess irregularity in images. In short, $SampEn_{2D}$ has two parameters: m and r. Parameter m is responsible for setting the window size that is used to search the matching patterns all over the matrix; parameter r is responsible for setting tolerance, i.e., when r is larger, the matching criteria become more permissive.

We generate a two-dimensional matrix using the cumulative travel distances to visualize the spatial variation over time. Assuming the email account's lifetime is N days, the matrix has a shape of N by 24, as illustrated in Fig. 7. If the entropy of the matrix is higher, it implies that the matrix's complexity is greater, indicating that the spatial variation is more common. From Fig. 7, we can observe that the account rarely changes its IP address during regular times, but it frequently changes its IP address for consecutive days. The corresponding matrix has a lower entropy value and a larger cumulative travel distance, suggesting that the account might have been compromised.

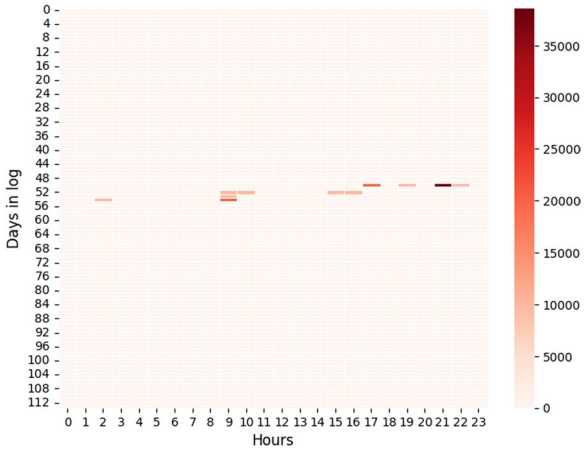

Fig. 7 Spatial characterization result of an email account in Dataset-1

Ranking

We introduce the scoring function $s(std, en)$ to evaluate email accounts' spatial behavior, where accounts with higher scores are more likely to be compromised. The function has two parameters: *std*, the standard deviation of the cumulative travel distance series, which measures the extremeness of the spatial variation, and *en*, the $SampEn_{2D}$ value of the matrix constructed as described above, which measures the commonness of the spatial variation. The function is fuzzable and only needs to ensure that the score is positively correlated with *std* and inversely correlated with *en*. Similarly, we could use two thresholds to divide the results into four areas, but this requires additional processing of the cumulative travel distances due to their wide range of values and lack of a reference standard. Once we score all email accounts, we can obtain a ranked list of accounts based on their likelihood of compromise.

Application architecture

When characterizing the logins' temporal behavior, we propose the concept of lifetime of subnet, so the framework does not need too many days of login logs as input. Meanwhile, when characterizing the logins' spatial behavior, the number of days is used as a dimension of the matrix, and there is no over-reliance on historical logs when calculating the $SampEn_{2D}$. Consequently, our framework would work fine for continuous input logs with a monthly or quarterly diagnosis window. In addition, we can use the framework to check for compromise immediately after a severe vulnerability is disclosed or an APT campaign is revealed.

The architecture of CEAD consists of a temporal behavior characterisation module and a spatial behavior characterisation module. In detecting compromised email accounts, CEAD has four stages: preparation, characterisation, determination, and diagnosis.

Preparation stage: In the preparation stage, we extract four basic features: protocol, datetime, IP address and email account, from the input login logs and store them in the database. Additionally, we need to establish an IP-Geo database to query geolocation according to IP address.

Characterization stage: In the characterization stage, we begin by assessing the reputation of the subnet based on the input logs and store the results in the database. Once the temporal behavior characterization module is completed, we obtain a corresponding result tuple (*w*, *rep*) for each subnet. Similarly, once the spatial behavior characterization module is completed, we obtain a corresponding result tuple (*std*, *en*) for each email account. Notably, we build reference models for each login protocol individually as email account owners may have distinct usage patterns when using different protocols (which could correspond to different clients). For instance, employees may use webmail during the day and mobile applications after work hours. To mitigate false positives for protocols that are used infrequently in characterizing spatial behavior, we construct the cumulative travel distance matrix for each protocol first and then aggregate them.

Determination stage: For the temporal characterization results, we set two thresholds to obtain an initial list of suspicious subnets from the suspicious area. Then, we filter the initial list both horizontally and vertically and output a subnet list, which is then sorted by their reputation. For the spatial characterization results, we use $s(std, en)$ to score each account and output a list of accounts sorted by their score. Next, we convert the obtained subnet list and account list into the same form and merge them. Specifically, for each subnet output by the temporal characterization module, we find all accounts that have been logged in to by that subnet and generate an ordered list $L1$, for example, $[(account_1, subnet_1), (account_2, subnet_1) \ldots (account_n, subnet_m)]$. Similarly, for each account output by the spatial characterization module, we find all suspicious subnets that have logged in to that account and generate an ordered list $L2$, for example, $[(account_1, subnet_1), (account_1, subnet_2) \ldots (account_p, subnet_q)]$. The determination of suspicious subnets is based on the following criteria: we first identify the hourly blocks with cumulative travel distances greater than the mean value of the cumulative travel distance matrix, and then locate the subnets that appear in these blocks, which are subsequently sorted by their reputation values. Finally, we merge $L2$ to $L1$ to produce a final list of account-subnet pairs.

Diagnosis stage: At this stage, security analysts can select a number of top-ranked account-subnet pairs for analysis based on their workload. If an account is indeed compromised, they can assess the damage immediately and identify the corresponding IP address as an IoC. Based on the accuracy of the analysis results, analysts can investigate the reasons and make adjustments, such as modifying the subnet size and threshold. Analysts can also fine-tune the ranking method to find a suitable configuration for their scenario.

Evaluation and analysis

In this section, we first introduce the datasets that we utilized and present the results of the data analysis using conventional methods. Next, we describe the configurations that we used to evaluate CEAD. Finally, we analyze the evaluation results, compare them with DAS, and based on our findings, reveal some previously undisclosed attack activities.

Datasets

Our dataset consists of 9 organizations' email logs. These organizations have different sizes and use the same email system hosted by a third party. Due to privacy concerns, the email service provider only provided login logs. Table 2 shows the basic statistics of our datasets. Dataset-1, Dataset-5, and the rest of the datasets contain login logs for 113 days, 70 days, and 277 days, respectively. As can be seen from Table 2, Web and SMTP have fewer logins than the other two protocols. In addition, the number of unique IP addresses and unique subnets may be positively correlated with the number of email accounts and the number of days of logs.

After we obtained these data, we conducted a preliminary investigation of the accounts' security status using standard methods, including (1) making statistics on changes in the login location, (2) counting the number of IP addresses that have logged in to each account and the number of accounts that have been logged in by each IP address, and (3) matching IP addresses according to IoCs.

Location-changed logins: We consider a login to be a location-changed login if two consecutive logins have a geographical distance of more than 100 KM. We then calculate the percentage of location-changed logins among all logins. As shown in Fig. 1, all datasets contain location-changed logins, with the highest percentage reaching nearly 25%. However, it is not reliable to evaluate an organization's email account security solely based on the proportion of location-changed logins, as this statistical characteristic may vary depending on the industry and work style of the organization. In almost all datasets, Web and SMTP have more location-changed logins than the other two protocols. We speculate that the reason for this phenomenon is that Web and SMTP are usually used when users log in on their own. In contrast, POP3 and IMAP are usually used when clients log in continuously and automatically.

The number of IP addresses that have logged in to each account, and the number of accounts that have been logged in by each IP address: Intuitively, if an account is accessed from too many IP addresses, it is more likely to be compromised. Similarly, if an IP address logs into too many accounts, it is considered suspicious. Figures 8 and 9 depict the statistical results of these indicators. The figures show that most email accounts have fewer than 100 login IP addresses, and only a small fraction of IP addresses have accessed more than 10 email accounts. However, setting fixed thresholds for these indicators to make a determination would introduce a lot of false positives due to the complex usage scenarios such as the use of DHCP or proxy pools.

Identification based on IoCs: When lacking support from other labels, it is common to utilize IoCs to match suspicious IP addresses. In our study, we employed the open-source threat intelligence Alienvault-OTX (AlienVault 2022) as a reference and considered the lag and timeliness of IoCs. We deemed an IP address malicious if its login time fell between the time when a related pulse was released or updated and half a year before the pulse was released. Regrettably, the initial matching results showed that no IP addresses were identified as malicious after matching all IP addresses in the nine datasets. This outcome highlights the constraints of relying exclusively on threat intelligence.

Therefore, we expanded the matching scope by using /24 subnet as the matching range. If there is a matching result for the same subnet IP address as the login IP address, the /24 subnet is considered a malicious subnet. The matching results are shown in Table 3, where the malicious subnets have been anonymized. When identifying malicious subnets, we ignore the matching results if the location of the malicious subnet is the common location of the email account or if the account owner has a habit of using proxies.

Table 2 Statistics for 9 datasets

	Unique accounts	Unique IP addrs	Unique subnets (/24)	Account-subnet pairs	Number of logins				
					Web	SMTP	POP3	IMAP	Total
Dateset-1	323	15,371	2993	8632	25,875	47,642	320,971	1,682,202	2,076,690
Dateset-2	543	29,619	7876	17,580	119,269	24,227	259,891	5,523,264	5,926,651
Dateset-3	1356	84,522	16,709	46,695	312,523	237,120	981,876	19,139,990	2,0671,509
Dateset-4	1040	38,793	8179	21,886	143,113	82,473	3,212,183	7,392,472	10,830,241
Dateset-5	2910	68,924	17,383	40,035	99,755	23,427	691,813	13,900,911	14,715,906
Dateset-6	923	44,435	9018	23,246	149,160	196,680	4,813,528	7,417,580	12,576,948
Dateset-7	1445	86,592	17,490	45,442	274,772	91,222	1,364,399	16,358,601	18,088,994
Dateset-8	1799	61,992	13,727	47,939	207,303	351,734	1,794,550	14,626,653	16,980,240
Dateset-9	745	61,078	12,747	31,563	123,679	24,894	652,664	6,543,363	7,344,600

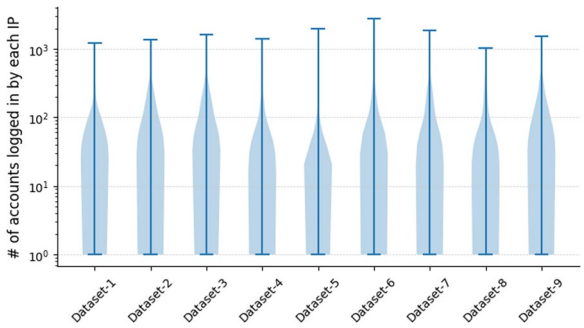

Fig. 8 The number of IP addresses that have logged in to each account in our datasets

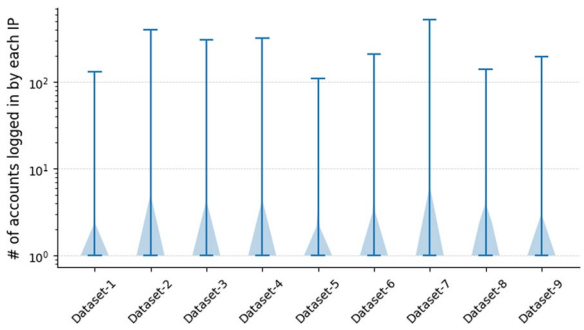

Fig. 9 The number of accounts that have been logged in by each IP address in our datasets

After manual analysis, we divide the final matching results into two categories: confirmed malicious subnets and uncertain ones. Confirmed malicious subnets are determined based on the existence of sufficient and obvious attack behavior features. If we cannot find sufficient evidence to determine a subnet as a malicious one or the login frequency of the subnet is too low, we mark it as uncertain. These results may be due to either the large matching scope or limited log data.

Table 3 IoCs matching results

Subnet	In dataset	Type	IP addrs	Associated accounts	Maximum login times
A	Dateset-1	Malicious	1	129	24
B	Dateset-1	Malicious	9	7	11485
C	Dateset-3	Uncertain	1	2	11
D	Dateset-3	Uncertain	1	1	14
E	Dateset-5	Uncertain	1	1	2
F	Dateset-5	Uncertain	1	1	9
G	Dateset-5	Uncertain	1	1	1
H	Dateset-5	Uncertain	1	1	1
I	Dateset-5	Uncertain	1	1	2
J	Dateset-5	Uncertain	1	1	1
K	Dateset-6	Malicious	1	55	4
L	Dateset-7	Uncertain	1	1	2
M	Dateset-7	Uncertain	1	1	9

As shown in Table 3, we identified suspicious subnets in five datasets after expanding the matching scope. Based on the statistics, Dataset-1 suffered more severe attacks and was targeted by at least two attack groups. Notably, Subnet A in Dataset-1 and subnet K in Dataset-6 both contained only one IP address, which was also the same IP address, and had a high number of logins in some email accounts in Dataset-1. This provided us with sufficient data to build the attacker's model. Therefore, if CEAD can identify the malicious subnet A in Dataset-1, we can identify the corresponding IP address as a new IoC, which provides us with the opportunity to discover the malicious IP address in Dataset-6, even if that IP address has a low login frequency in Dataset-6.

Configuration

Our framework needs to specify some parameters, calculations and rules for detection, including (1) parameters and data selection of mixture model; (2) parameters of $SampEn_{2D}$; (3) measure function for subnet reputation; and (4) anomaly determination rules.

Parameters and data selection of mixture model: In our evaluation, we configure the following parameters to build a mixture model: (1) K, we set $K = 11$, i.e., we use one reference model and 10 Gaussian models to fit the data; (2) ωs, we set the initial weight of the reference model to 0.99 and 10 Gaussian models share the remaining 0.01 equally; (3) μs, we set different μs for the 10 Gaussian models, and these μs divide 0–86,400 into 11 equal segments; (4) σs, we set the σs of all Gaussian models to 20,000; (5) data, for logins during the lifetime of a newly-appeared-subnet, we use login data of subnets with more logins than the newly-appeared-subnet to build the reference model, and use the average of the reputations of these subnets as the reference model's reputation. Additionally, we ignore subnets with fewer logins than the newly-appeared-subnet, that is, we use 10 Gaussian models to fit the distribution of the newly-appeared-subnet. If the newly-appeared-subnet has the most logins during its lifetime, we mark it as *max*, and if it has less than one day of life or has less than 10 logins, we mark it as *ne*.

Parameters of $SampEn_{2D}$: m and r are two necessary parameters of $SampEn_{2D}$, representing the window size and the tolerance, respectively. In our evaluation, we use the default values of m (2×2) and r ($0.2 \times std$).

Measure function for subnet reputation: As previously mentioned, we employ $Feature_A$, $Feature_B$, and $Feature_C$ to assess the reputation of one subnet. We defined the calculation of the subnet's reputation as:

$$rep = \log(Feature_C \times (Feature_A + Feature_B)) \quad (7)$$

Determination rules: We use two thresholds to outline trusted and suspicious areas. In our evaluation, we using different combinations of w' and rep' to summarize the threshold selection guidelines. Specifically, when rep' was set to the lower bound of the top 30% of all reputation values, we set w' to 0.85, 0.90, and 0.95. When w' was set to 0.90, we set rep' to the lower bound of the top 10%, 30%, and 50% of all reputation values.

We score the results of the spatial behavior by $score = std/en$. The email accounts with higher scores are more likely compromised.

Detection result

We evaluated our framework on an Ubuntu 18.04 server with 32 cores and 64 GB of memory. However, without ground truth, we are unable to confirm the account-subnet pairs ranked high by CEAD's output. Therefore, we use the IoC matching results in Table 3 as a reference.

We first analyzed the detection results of the temporal behavior characterization module and the spatial behavior characterization module separately, and then compared the final merged results of CEAD with DAS. In addition, we manually analyzed several account-subnet pairs with high ranking output by CEAD, in an attempt to discover undisclosed attack activities.

Temporal behavior characterization result

Figure 6 shows the results of temporal behavior characterization on the nine datasets. Based on these results, we select different combinations of thresholds for w' and rep' to calculate the detection rate, as shown in Tables 4 and 5.

Table 4 Detection results of the temporal behavior characterization module for different values of w', with rep' fixed

Dataset	w'	Top % of all reputation values (%)	rep'	Subnets in suspicious area	Subnets after vertical filtering	Subnets after horizontal filtering (% decrease from the total count)	Associated accounts	Detection rate
Dataset-1	0.85	30	−1.83	100	49	22(99.26%)	31	15/132
Dataset-2	0.85	30	−1.87	203	127	78(99.01%)	100	–
Dataset-3	0.85	30	−1.79	649	259	106(99.37%)	135	0/3
Dataset-4	0.85	30	−1.93	529	275	107(98.69%)	142	–
Dataset-5	0.85	30	−1.27	427	240	141(99.19%)	217	1/6
Dataset-6	0.85	30	−1.69	420	216	77(99.15%)	109	4/55
Dataset-7	0.85	30	−1.86	676	300	102(99.42%)	136	1/2
Dataset-8	0.85	30	−1.79	679	302	150(98.91%)	193	–
Dataset-9	0.85	30	−2.06	164	68	35(99.73%)	94	–
Dataset-1	0.90	30	−1.83	134	65	31(98.96%)	153	130/132
Dataset-2	0.90	30	−1.87	265	168	100(98.73%)	117	–
Dataset-3	0.90	30	−1.79	790	346	139(99.17%)	184	0/3
Dataset-4	0.90	30	−1.93	641	360	142(98.26%)	181	–
Dataset-5	0.90	30	−1.27	534	318	194(98.88%)	299	1/6
Dataset-6	0.90	30	−1.69	517	288	117(98.70%)	154	7/55
Dataset-7	0.90	30	−1.86	834	394	145(99.17%)	178	1/2
Dataset-8	0.90	30	−1.79	825	399	197(98.56%)	248	–
Dataset-9	0.90	30	−2.06	243	105	62(99.51%)	119	–
Dataset-1	0.95	30	−1.83	178	91	43(98.56%)	170	130/132
Dataset-2	0.95	30	−1.87	366	255	146(98.15%)	148	–
Dataset-3	0.95	30	−1.79	1000	495	188(98.87%)	223	0/3
Dataset-4	0.95	30	−1.93	790	487	207(97.47%)	231	–
Dataset-5	0.95	30	−1.27	705	438	275(98.42%)	379	1/6
Dataset-6	0.95	30	−1.69	666	399	188(97.92%)	210	13/55
Dataset-7	0.95	30	−1.86	1053	544	215(98.77%)	236	1/2
Dataset-8	0.95	30	−1.79	1022	527	265(98.07%)	362	–
Dataset-9	0.95	30	−2.06	372	195	105(99.18%)	153	–

Table 5 Detection results of the temporal behavior characterization module for different values of *rep*′, with *w*′ fixed

Dataset	*w*′	Top % of all reputation values (%)	*rep*′	Subnets in suspicious area	Subnets after vertical filtering	Subnets after horizontal filtering (% decrease from the total count)	Associated accounts	Detection rate
Dataset-1	0.90	10	−0.56	61	50	30(99.00%)	187	131/132
Dataset-2	0.90	10	−0.42	194	158	120(98.48%)	193	–
Dataset-3	0.90	10	−0.28	436	336	192(98.85%)	464	2/3
Dataset-4	0.90	10	−0.44	461	336	184(97.75%)	317	–
Dataset-5	0.90	10	−0.11	262	213	176(98.99%)	564	1/6
Dataset-6	0.90	10	−0.25	292	234	148(98.36%)	283	11/55
Dataset-7	0.90	10	−0.31	513	380	174(99.01%)	285	1/2
Dataset-8	0.90	10	−0.28	626	481	306(97.77%)	507	–
Dataset-9	0.90	10	−0.50	125	78	32(99.75%)	121	–
Dataset-1	0.90	30	−1.83	134	65	31(98.96%)	153	130/132
Dataset-2	0.90	30	−1.87	265	168	100(98.73%)	117	–
Dataset-3	0.90	30	−1.79	790	346	139(99.17%)	184	0/3
Dataset-4	0.90	30	−1.93	641	360	142(98.26%)	181	–
Dataset-5	0.90	30	−1.27	534	318	194(98.88%)	299	1/6
Dataset-6	0.90	30	−1.69	517	288	117(98.70%)	154	7/55
Dataset-7	0.90	30	−1.86	834	394	145(99.17%)	178	1/2
Dataset-8	0.90	30	−1.79	825	399	197(98.56%)	248	–
Dataset-9	0.90	30	−2.06	243	105	62(99.51%)	119	–
Dataset-1	0.90	50	−3.11	166	74	36(98.80%)	152	130/132
Dataset-2	0.90	50	−3.31	306	188	96(98.78%)	128	–
Dataset-3	0.90	50	−3.29	1012	393	164(99.02%)	177	0/3
Dataset-4	0.90	50	−3.43	741	356	133(98.37%)	165	–
Dataset-5	0.90	50	−2.42	811	417	234(98.65%)	273	0/6
Dataset-6	0.90	50	−3.13	617	302	106(98.82%)	126	6/55
Dataset-7	0.90	50	−3.42	1057	436	181(98.97%)	203	1/2
Dataset-8	0.90	50	−3.30	941	400	184(98.66%)	230	–
Dataset-9	0.90	50	−3.62	349	141	86(99.33%)	82	–

Table 4 shows the effect of different values of *w*′ on the results when *rep*′ is fixed. It can be observed that as the threshold for *w*′ increases, the similarity judgement becomes more stringent, resulting in more suspicious subnets and associated email accounts. This increases the probability of successfully identifying compromised email accounts but also places a greater workload on the analysis team.

Table 5 shows the effect of different values of *rep*′ on the results when *w*′ is fixed. The reputation threshold mainly affects the initial output subnet count and filtering strength. A lower reputation threshold yields more initial subnets but also means more subnets will be filtered in the future. If analysts have a good understanding of subnet division and are confident in the subnet reputation, they should lower the reputation threshold, and vice versa.

Overall, CEAD's temporal behavior characterization module significantly reduced the number of suspicious subnets that require investigation compared to the original number of subnets, greatly reducing the workload of analysts and successfully detecting the confirmed malicious subnet A. For other suspicious subnets that were not detected, we conducted manual analysis and summarized the reasons into three points:

- The distribution of the subnet is similar to the reference model, that is, there is no obvious time zone difference between them, resulting in the subnet being classified into the trusted area. Subnets B, C, and D belong to this category. We will show in later sections that these subnets can be detected through the spatial behavior characterization module.
- The subnet has too few login events to build a model. Subnets E, F, G, H, I, J, L, and M belong to this category. Although some of the compromised email addresses were identified in the results, they were not detected through these targeting subnets. We

will investigate these compromised email accounts in later sections to determine whether their associated subnets belong to undisclosed malicious subnets.

- Although the subnet has a small number of logins and cannot build a model, it can be identified through intelligence sharing. Malicious subnet K belongs to this category.

Comparison with Jensen–Shannon Divergence The CEAD's temporal behavior characterization module utilizes the weight of the reference model to measure the similarity between the reference model and the mixture model. To demonstrate the superiority of our method, we conducted additional experiments. In these experiments, we used Jensen–Shannon divergence to measure the similarity between distributions and compared them in two ways: (1) by comparing the similarity between the reference model and the newly-appeared-subnet's model, and (2) by comparing the similarity between the reference model and the mixture model.

Figure 10 illustrates the number of subnets with similarity lower than a certain threshold for the three methods. ω-RM-MM represents the similarity between the reference model and the mixture model, calculated by our approach; js-RM-MM represents the similarity between the reference model and the mixture model, calculated by Jensen–Shannon divergence; js-RM-Subnet represents the similarity between the reference model and the distribution of a newly-appeared-subnet, calculated by Jensen–Shannon divergence.

Since the more similar the distributions, the smaller the value of the Jensen–Shannon divergence. Thus, we use the value of 1 − *Jensen–Shannon divergence* in the figure to be able to compare. The results in Fig. 10 are only shown for Dataset-1, and the results on other datasets are similar.

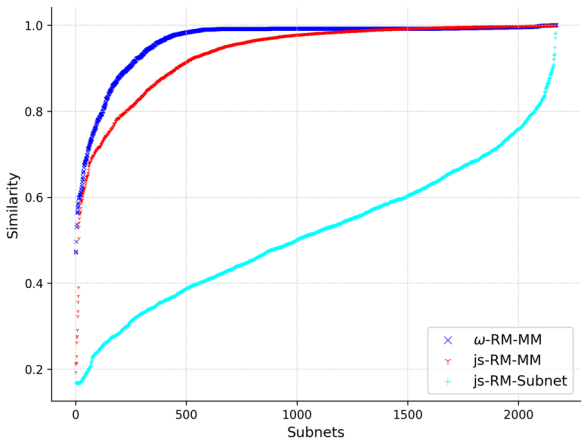

Fig. 10 The results of using three different similarity comparison methods on Dataset-1

Based on the results, the majority of reference model and newly-appeared-subnet models are dissimilar. This is mainly because in most cases, the data used to build the reference model and the newly-appeared-subnet model are imbalanced and may differ greatly, resulting in treating noise data as an independent distribution for comparison. In addition, the approximate linear results will pose a challenge in selecting the threshold. Even if we set a low threshold, this method will still output many subnets for analysis. Therefore, this similarity comparison method should not be used in practical implementation. It should be noted that the reason we are able to use Jensen–Shannon divergence for both horizontal and vertical filtering is that our objective is to identify subnets with similarity above a certain threshold (similar to the nonlinear part of the results in the upper right corner of Fig. 10), rather than those with dissimilarity below the threshold.

Comparing the similarity between the reference model and mixture model can avoid the aforementioned issue of data imbalance. This is because, after incorporating the data for constructing the newly-appeared-subnet's model, the data sizes for the reference model and mixture model are not significantly different in most cases, making the comparison more reasonable.

Based on Fig. 10, the performance of ω-RM-MM and js-RM-MM is almost identical when the similarity between the reference model and mixture model is very high or low. However, when the similarity falls within other ranges, ω-RM-MM is more discriminative. In other words, if we need to set a threshold to obtain a list of suspicious subnets, using the former method will yield fewer subnets.

Spatial behavior characterization result

Figure 11 shows the results of spatial behavior characterization on 9 datasets. Based on these results, we scored and ranked email accounts, and Table 6 shows the detection rate of compromised email accounts in the top 10%, 20%, and 30% of ranked accounts by the spatial behavior characterization module.

From Table 6, it can be seen that when the top 30% ranked email accounts are taken as the analysis object, almost all compromised email accounts are detected, except for some compromised email accounts associated with malicious subnet A and K in Dataset-1 and Dataset-6. Especially for the malicious subnet B, among the 7 email accounts that were logged in by it, 3 of them are ranked in the top 10. For the malicious subnet A and K, the time zone difference between the subnet's activity time and the owner's activity time, or the fact that there are only a few login events in some email accounts, led to

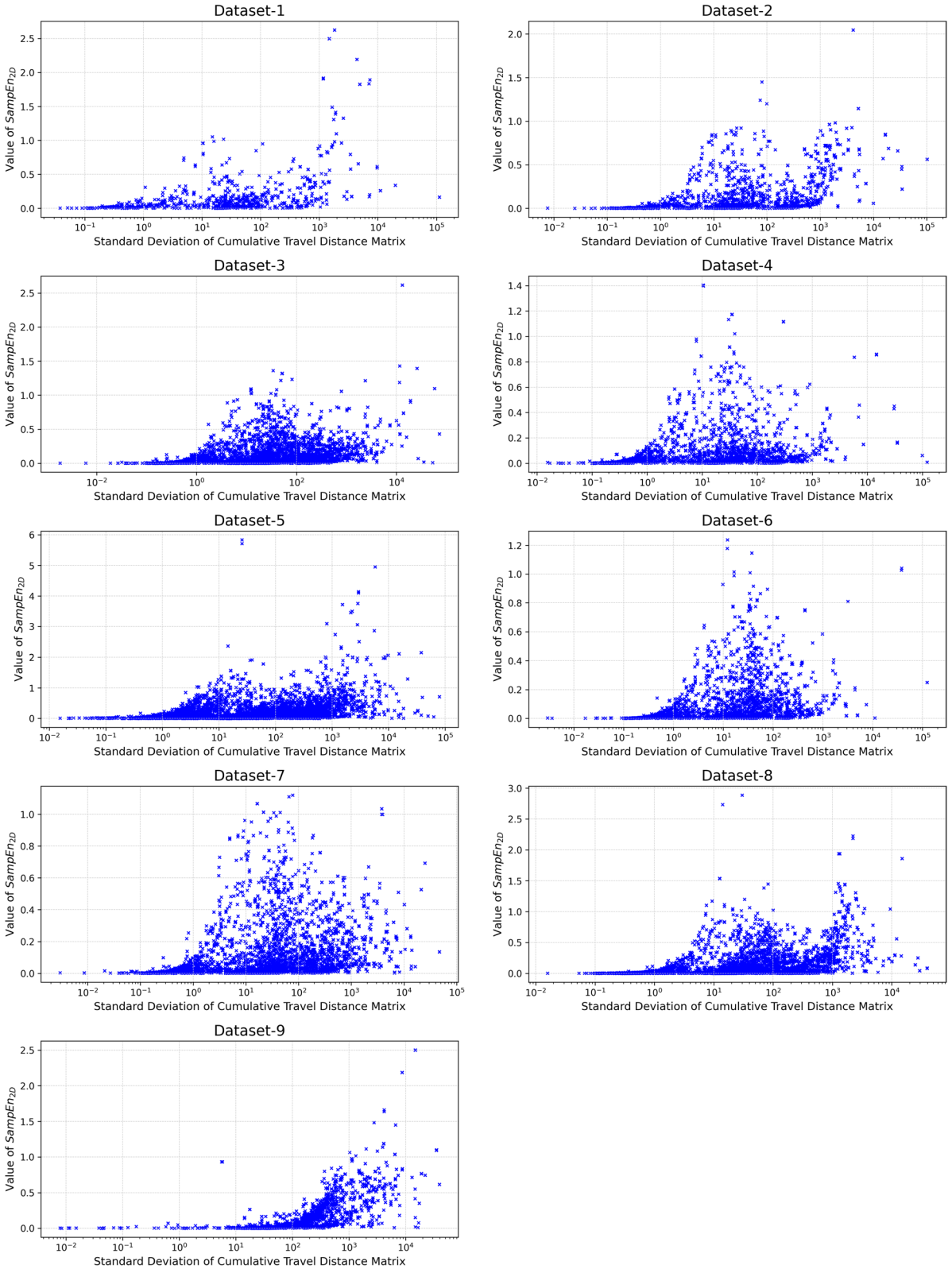

Fig. 11 Spatial characterization result of 9 datasets

Table 6 Detection results of the spatial behavior characterization module

Dataset	Top 10%		Top 20%		Top 30%	
	Emails accounts	Detection rate	Emails accounts	Detection rate	Emails accounts	Detection rate
Dataset-1	29	8/132	58	27/132	87	39/132
Dataset-2	46	–	93	-	140	–
Dataset-3	125	2/3	250	3/3	376	3/3
Dataset-4	82	–	165	–	247	–
Dataset-5	252	3/6	505	4/6	758	5/6
Dataset-6	73	11/55	147	20/55	220	28/55
Dataset-7	121	2/2	243	2/2	365	2/2
Dataset-8	150	–	301	–	452	–
Dataset-9	65	–	131	–	197	–

the lack of obvious alternating login behaviors, which was not reflected in the cumulative travel distance matrix as an anomaly. For the compromised email accounts associated with the other subnets in Table 3, even though the number of logins from the identified suspicious subnets was relatively small, CEAD's spatial behavior analysis module successfully detected these accounts. Therefore, we speculate that the attackers may have used other IP addresses that have not been identified as IoCs, which ultimately led to anomalous login series being detected by our method.

Comparison with DAS

We compared the performance of CEAD and DAS (Ho et al. 2017) when using only login logs. DAS extracts two features from login logs when a login event $E = (account', IP\ address')$ occurs: (1) the number of users who have logged in from the city where $IP\ address'$ is located before this event, and (2) the number of times $account'$ has logged in from that city before this event.

We made some modifications to DAS to make it applicable to our scenario:

- We replaced all IP addresses used in feature extraction in DAS with the /24 subnet to which the IP address belongs.
- During the scoring and ranking process, we only used the features of the first occurrence of the account-subnet pair for calculation, rather than for all login events. There are three main reasons for this: (1)when using only login logs, the two feature values of a later login event for the same account-subnet pair will always be greater than or equal to those of the first event, leading to a lower ranking for the later event; (2)as DAS' features are based on statistical analysis of historical data, the features of the first occurrence of an account-subnet pair (a login from an unseen IP address) are most likely to indicate the likelihood of an email account being compromised; (3)DAS ranks N objects with a calculation complexity of N^2, thus ranking account-subnet pairs rather than all login events significantly reduces computational costs.
- As DAS requires training data, we selected the first eighth of the data in each dataset as the training data for DAS.

After making the above modifications to DAS, both CEAD and DAS output rankings of account-subnet pairs. To compare which ranking is more effective, or more helpful for analysts, we designed a comparison method. We select 10%, 20%, and 30% of the total number of account-subnet pairs in each dataset as the workload threshold for analysts. For the rankings provided by CEAD and DAS, we count the number of compromised email accounts included within each workload threshold and use the hit rate as the evaluation metric. The results of the comparative experiments are shown in Table 7. In the comparative experiment, the ω' of CEAD's temporal characterization module is set to 0.9, and rep' is set to the lower bound value of all top 30% reputations.

From Table 7, it can be observed that when the workload threshold is low, CEAD outperforms or is comparable to DAS in all datasets except Dataset-6. However, when the workload threshold is high, DAS detects more compromised email accounts. This suggests that CEAD is better at ranking potentially compromised email accounts higher, but its detection results are not as comprehensive as those of DAS. CEAD's performance on Dataset-6 is worse than DAS, because the login frequency of associated malicious subnets is extremely low, making it difficult for our method to model them,

Table 7 Detection rate of CEAD and DAS at different workload thresholds

Dataset	Total pairs	Top 10%			Top 20%			Top 30%		
		Pairs to be analyzed	CEAD	DAS	Pairs to be analyzed	CEAD	DAS	Pairs to be analyzed	CEAD	DAS
Dataset-1	8632	863	**132/136**	20/136	1726	**134/136**	93/136	2589	135/136	**136/136**
Dataset-2	17580	1758	–	–	3516	–	–	5274	–	–
Dataset-3	46695	4669	**1/3**	0/3	9339	3/3	3/3	14008	3/3	3/3
Dataset-4	21886	2188	–	–	4377	–	–	6565	–	–
Dataset-5	40035	4003	3/6	3/6	8007	4/6	**6/6**	12010	5/6	**6/6**
Dataset-6	23246	2324	7/55	**19/55**	4649	13/55	**55/55**	6973	20/55	**55/55**
Dataset-7	45442	4544	**2/2**	0/2	9088	**2/2**	1/2	13632	**2/2**	1/2
Dataset-8	47939	4793	–	–	9587	–	–	14381	–	–
Dataset-9	31563	3156	–	–	6312	–	–	9468	–	–

The bold in the table is to highlight the system that perform better

and the too few login behaviors did not cause a significant increase in the cumulative travel distance. However, if we consider intelligence sharing, CEAD's detection performance will be improved.

Additionally, the selection of training data has a significant impact on the detection results of DAS. We compared the detection rate of CEAD and DAS with different learn sizes when the workload threshold is 20%, and the results are shown in Table 8. It can be observed that using too little or too much training data in DAS cannot achieve satisfactory results. Too little training data can result in normal account-subnet pairs being ranked higher, while too much training data may include abnormal account-subnet pairs in the training data. In contrast, CEAD uses the concept of subnet lifetime and cumulative travel distance matrix, and does not overly rely on the size of the logs or require additional training data.

Undisclosed malicious subnet detection results

After analyzing a portion of the account-subnet pairs in the ranking output of CEAD, we found some highly suspicious login behaviors. We hereby list some subnets and accounts that exhibit clear abnormal behavior and do not match any IoCs as follows:

- In Dataset-1, there exists a malicious subnet N which is ranked 17th by the temporal characterization module. This subnet contains only one IP address and has logged in to three email accounts. The login methods, activity dates, and distribution of this subnet across the 3 email accounts are all very similar, and the city where the IP address is located is not a usual location for any of the accounts. Therefore, we infer that this malicious subnet is an infrastructure used by an attack organization in a single attack campaign.

Table 8 Detection rate of CEAD and DAS with different learn sizes

Dataset	Total pairs	Pairs to be analyzed	CEAD	DAS with different learn sizes			
				0	1/8	1/4	1/2
Dataset-1	8632	1726	134/136	39/136	93/136	117/136	56/136
Dataset-2	17580	3516	–	–	–	–	–
Dataset-3	46695	9339	3/3	1/3	3/3	3/3	3/3
Dataset-4	21886	4377	–	–	–	–	–
Dataset-5	40035	8007	4/6	4/6	6/6	6/6	6/6
Dataset-6	23246	4649	13/55	55/55	55/55	55/55	0/55
Dataset-7	45442	9088	2/2	1/2	1/2	1/2	1/2
Dataset-8	47939	9587	–	–	–	–	–
Dataset-9	31563	6312	–	–	–	–	–

- In Dataset-1, there are anomalies in email accounts ranked third and fourth in the spatial behavior characterization results. These two email accounts do not have a habit of using proxies, but their occasional logins from faraway locations resulted in an abnormal cumulative travel distance. The malicious subnet associated with the first email account also logged in to four other email accounts, and the login frequency in these accounts was very low. We speculate that this is the attacker's attempt to verify leaked account information. The malicious subnet associated with the second email account also logged into 15 other email accounts within three hours of a day, with each account being logged in only 1–2 times. Their activity dates and login methods are similar to those of malicious subnet A, so we speculate that they belong to the same attack organization.
- In Dataset-6, there is an anomaly in the spatial behavior characterization result of the 13th ranked email account. The account does not exhibit any proxy usage habit, and it is associated with two malicious subnets that both perform password verification. One of the subnets has logged in to 19 email accounts, and the other has logged in to 26 email accounts. The behavior of one of the subnets is similar to the one observed in malicious subnet A, suggesting that they belong to the same attacking organization.

Based on discovered real attack cases, we summarized several characteristics of APT groups when stealing emails. First, the same attacker uses changing IP addresses to carry out attacks. For example, the attack organization corresponding to the malicious subnet A used IP addresses from at least 3 different subnets to access the victim's email account. Second, attackers adopt different strategies for different targets. For high-value targets, attackers will track for a long time, while for most other targets, they rarely log in after verifying their access permissions. Third, the number of email theft attacks far exceeds lateral phishing attacks. Fourth, attackers use a combination of manual and automated attack methods. They first verify account permissions manually and filter out high-value targets, and then use automated tools to steal emails for a long time. This is mainly reflected in the fact that the malicious subnet used one protocol for the first login and another protocol for subsequent logins.

Discussion and limitations

In this section, we summarize some limitations of our framework, and briefly describe our framework's efficiency and extensibility.

Limitations

The evaluation results demonstrate that our approach can (1) effectively ranking the suspicious account-subnet pairs towards the top and (2) identifying previously undisclosed malicious IP addresses. However, our approach does have some limitations in the following areas.

Application scenario: First, our approach detects compromised email accounts by identifying login behaviors that do not belong to the owner. However, in actual enterprise settings, some accounts are used by multiple users, which may lead to false positives. Second, our approach is designed to support attribution and forensic efforts, and may not be easily retrofitted for real-time detection methods. Third, our approach primarily focuses on detecting compromised email accounts within enterprise settings. It may not be able to detect fake accounts that are maliciously registered by adversaries through automated means, as they may not exhibit normal behaviors for comparison.

Sparse logins and subnet size: The evaluation of our approach revealed that it may not be effective in detecting malicious subnets with sparse logins. If attackers utilize frequently changing IP addresses, such as through Tor or proxy pools, to target different accounts, our framework may not be capable of detecting such activities. Furthermore, the size of the subnet also plays a significant role in the accuracy of detection results. An unreasonable subnet size may lead to inaccurate reputation computations and ultimately produce incorrect characterization results. In contrast, by applying our framework with a more reasonable subnet size reference, security teams may achieve better results.

Evasion strategies: Adversaries may alter their strategies to evade detection by our approach. For instance, attackers may utilize frequently changing IP addresses or conceal their time zone, and choose IP addresses in close proximity to the target to carry out their malicious activities. Such tactics can prevent the attacker's logins from being considered as independent behavior or being dismissed as noise by our method.

Efficiency and extensibility

We evaluated our framework in an environment without GPUs, which resulted in significant time consumption (nearly 10 h for characterizing the temporal behaviors in Dataset-3). However, if GPUs were utilized, the performance could be improved by several hundred times (Machlica et al. 2011).

Our method can extend the analysis object to more types of events after appropriate modification, such as posting

events and liking events in social network behavior analysis. Therefore, the types of compromised accounts that our framework can detect can also be extended to other types of accounts. In that scenario, $Feature_C$ for measuring the subnet's reputation can be calculated by the number of User-Agents (UAs) or device types.

Conclusion

We propose a framework for detecting compromised email accounts that relies solely on login data and does not require human labeling. Our approach identifies login behaviors that do not belong to the account owner by characterizing temporal and spatial patterns of logins. The detection results in nine datasets show that in some actual attacks, the attacker and the owner indeed have different time zones and alternate login behaviors. Our approach successfully detects malicious subnets and compromised accounts, and is more efficient than similar studies at low workload thresholds. Additionally, our method also has the ability to identify undisclosed malicious IP addresses.

Acknowledgements
The authors of the paper sincerely appreciate anonymous reviewers whoreviewed this manuscript and provided constructive comments.

Author contributions
JZ participated in all the work, designed the framework and experiments, and wrote the manuscript. CY and YC modified and replicated the DAS algorithm in a comparative experiment. DW and YL provided suggestions on the framework design and joined the discussion of this work. XC and QL reviewed the manuscript and gave suggestions on the revision of the details of the article. All authors read and approved the final manuscript.

Funding
This work is supported by the Youth Innovation Promotion Association CAS (No.2019163), the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDC02040100), the Key Laboratory of Network Assessment Technology at Chinese Academy of Sciences and Beijing Key Laboratory of Network security and Protection Technology.

Availability of data and materials
The experimental dataset used in this study includes user email addresses, login IP addresses, active times, etc. Due to privacy protection considerations, the involved dataset will not be made public.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 6 April 2023 Accepted: 11 June 2023
Published online: 04 September 2023

References

AlienVault (2022) Open Threat Exchange. https://otx.alienvault.com/browse/global/pulses
CNN (2021) Fake FBI emails about a sophisticated attack are part of 'ongoing situation,' agency says. https://edition.cnn.com/2021/11/13/politics/fbi-fake-emails-cyber-threat/index.html
Corp AS (2022) H2 2022 email threat report. https://abnormalsecurity.com/resources/h2-2022-report-brand-impersonation-phishing
Egele M, Stringhini G, Kruegel C, Vigna G (2013) Compa: detecting compromised accounts on social networks. In: NDSS
Egele M, Stringhini G, Kruegel C, Vigna G (2015) Towards detecting compromised accounts on social networks. IEEE Trans Dependable Secure Comput 14(4):447–460
Hao S, Syed NA, Feamster N, Gray AG, Krasser S (2009) Detecting spammers with snare: spatio-temporal network-level automatic reputation engine. In: USENIX security symposium, vol 9
Ho G, Sharma A, Javed M, Paxson V, Wagner D (2017) Detecting credential spearphishing in enterprise settings. In: 26th USENIX security symposium (USENIX security 17), pp 469–485
Ho G, Cidon A, Gavish L, Schweighauser M, Paxson V, Savage S, Voelker GM, Wagner D (2019) Detecting and characterizing lateral phishing at scale. In: 28th USENIX security symposium (USENIX security 19), pp 1273–1290
Hu X, Li B, Zhang Y, Zhou C, Ma H (2016) Detecting compromised email accounts from the perspective of graph topology. In: Proceedings of the 11th international conference on future internet technologies, pp 76–82
ipgeolocation (2022) Free IP geolocation API and accurate IP geolocation database. https://ipgeolocation.io
Karimi H, VanDam C, Ye L, Tang J (2018) End-to-end compromised account detection. In: 2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). IEEE, pp 314–321
Machlica L, Vanek J, Zajic Z (2011) Fast estimation of gaussian mixture model parameters on GPU using CUDA. In: 2011 12th international conference on parallel and distributed computing, applications and technologies. IEEE, pp 167–172
MaxMind (2022) Geolocate an IP address using Web Services. https://dev.maxmind.com/geoip/geolocate-an-ip/web-services?lang=en
Nur AY, Tozal ME (2018) Identifying critical autonomous systems in the internet. J Supercomput 74(10):4965–4985
Pv S, Bhanu S (2020) UbCadet: detection of compromised accounts in twitter based on user behavioural profiling. Multimed Tools Appl 79(27):19349–19385
Reynolds DA (2009) Gaussian mixture models. Encycl Biom 741:659–663
Ruan X, Wu Z, Wang H, Jajodia S (2015) Profiling online social behaviors for compromised account detection. IEEE Trans Inf Forensics Secur 11(1):176–187
Silva LEV, Senra Filho A, Fazan VPS, Felipe JC, Junior LM (2016) Two-dimensional sample entropy: assessing image texture through irregularity. Biomed Phys Eng Express 2(4):045002
Stringhini G, Mourlanne P, Jacob G, Egele M, Kruegel C, Vigna G (2015) {EVILCOHORT}: detecting communities of malicious accounts on online services. In: 24th USENIX security symposium (USENIX security 15), pp 563–578
UpGuard (2022) The 67 biggest data breaches. https://www.upguard.com/blog/biggest-data-breaches
Velayudhan SP, Somasundaram MSB (2019) Compromised account detection in online social networks: a survey. Concurr Comput Pract Exp 31(20):5346
Viswanath B, Bashir MA, Crovella M, Guha S, Gummadi KP, Krishnamurthy B, Mislove A (2014) Towards detecting anomalous user behavior in online social networks. In: 23rd USENIX security symposium (USENIX security 14), pp 223–238
Wikipedia (2022a) Advanced persistent threat. https://en.wikipedia.org/wiki/Advanced_persistent_threat
Wikipedia (2022b) Hillary Clinton email controversy. https://en.wikipedia.org/wiki/Hillary_Clinton_email_controversy
Wikipedia (2022c) Kernel density estimation. https://en.wikipedia.org/wiki/Kernel_density_estimation
Wikipedia (2022d) Jensen–Shannon divergence. https://en.wikipedia.org/wiki/Jensen-Shannon_divergence
Xu L, Jordan MI (1996) On convergence properties of the EM algorithm for gaussian mixtures. Neural Comput 8(1):129–151

Publisher's Note