(12) STANDARD PATENT (11) Application No. **AU 2019403265 B2**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Threat detection platforms for detecting, characterizing, and remediating email-based threats in real time

(51) International Patent Classification(s)
G06F 21/55 (2013.01) ***G06N 20/00*** (2019.01)
G06F 21/50 (2013.01)

(21) Application No: **2019403265** (22) Date of Filing: **2019.12.18**

(87) WIPO No: **WO20/132137**

(30) Priority Data

(31) Number	(32) Date	(33) Country
16/672,854	**2019.11.04**	**US**
62/807,888	**2019.02.20**	**US**
62/782,158	**2018.12.19**	**US**
62/813,603	**2019.03.04**	**US**

(43) Publication Date: **2020.06.25**
(44) Accepted Journal Date: **2023.05.18**

(71) Applicant(s)
Abnormal Security Corporation

(72) Inventor(s)
JEYAKUMAR, Sanjay;BRATMAN, Jeshua;CHECHIK, Dmitry;BAGRI, Abhijit;REISER, Evan;LIAO, Sanny Xiao Yang;LEE, Yu Zhou;GASPERI, Carlos Daniel;LAU, Kevin;JIANG, Kai Jing;TAN, Su Li Debbie;KAO, Jeremy;YEH, Cheng-Lin

(74) Agent / Attorney
Davies Collison Cave Pty Ltd, Level 15 1 Nicholson Street, MELBOURNE, VIC, 3000, AU

(56) Related Art
US 10104029 B1
US 9774626 B1
US 7610344 B2
US 2018/0091476 A1
US 2017/0237776 A1
US 2016/0014151 A1

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 June 2020 (25.06.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/132137 A1

(51) International Patent Classification:
G06F 21/55 (2013.01) *G06N 20/00* (2019.01)
G06F 21/50 (2013.01) *H04L 29/02* (2006.01)

(21) International Application Number:
PCT/US2019/067279

(22) International Filing Date:
18 December 2019 (18.12.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

62/782,158	19 December 2018 (19.12.2018)	US
62/807,888	20 February 2019 (20.02.2019)	US
62/813,603	04 March 2019 (04.03.2019)	US
16/672,854	04 November 2019 (04.11.2019)	US

(71) Applicant: **ABNORMAL SECURITY CORPORATION** [US/US]; 797 Bryant Street, San Francisco, CA 94107 (US).

(72) Inventors: **JEYAKUMAR, Sanjay**; 797 Bryant Street, San Francisco, CA 94107 (US). **BRATMAN, Jeshua**; 797 Bryant Street, San Francisco, CA 94107 (US). **CHECHIK, Dmitry**; 797 Bryant Street, San Francisco, CA 94107 (US). **BAGRI, Abhijit**; 797 Bryant Street, San Francisco, CA 94107 (US). **REISER, Evan**; 797 Bryant Street, San Francisco, CA 94107 (US). **LIAO, Sanny Xiao Yang**; 797 Bryant Street, San Francisco, CA 94107 (US). **LEE, Yu, Zhou**; 797 Bryant Street, San Francisco, CA 94107 (US). **GASPERI, Carlos, Daniel**; 797 Bryant Street, San Francisco, CA 94107 (US). **LAU, Kevin**; 797 Bryant Street, San Francisco, CA 94107 (US). **JIANG, Kai, Jing**; 797 Bryant Street, San Francisco, CA 94107 (US). **TAN, Su, Li Debbie**; 797 Bryant Street, San Francisco, CA 94107 (US). **KAO, Jeremy**; 797 Bryant Street, San Francisco, CA 94107

(54) Title: THREAT DETECTION PLATFORMS FOR DETECTING, CHARACTERIZING, AND REMEDIATING EMAIL-BASED THREATS IN REAL TIME

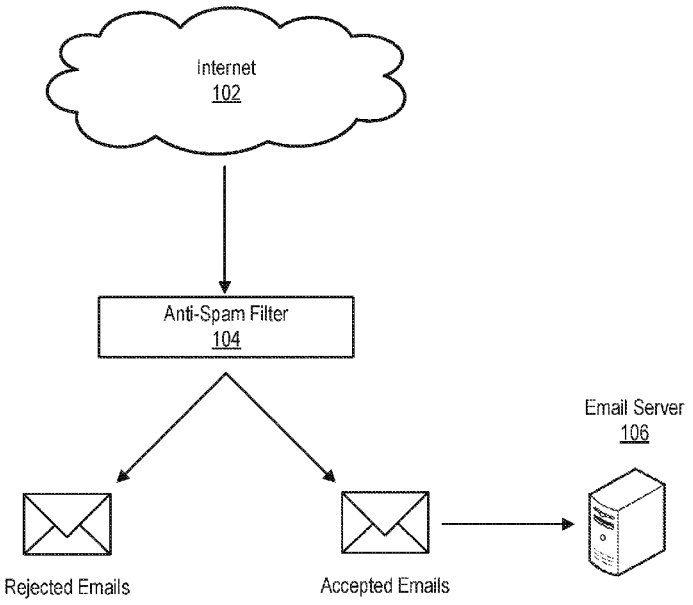

FIG. 1

(57) **Abstract:** Conventional email filtering services are not suitable for recognizing sophisticated malicious emails, and therefore may allow sophisticated malicious emails to reach inboxes by mistake. Introduced here are threat detection platforms designed to take an integrative approach to detecting security threats. For example, after receiving input indicative of an approval from an individual to access past email received by employees of an enterprise, a threat detection platform can download past emails to build a machine learning (ML) model that understands the norms of communication with internal contacts (e.g., other employees) and/or external contacts (e.g., vendors). By applying the ML model to incoming email, the threat detection platform can identify security threats in real time in a targeted manner.

WO 2020/132137 A1

[Continued on next page]

(US). **YEH, Cheng-Lin**; 797 Bryant Street, San Francisco, CA 94107 (US).

(74) **Agent: PETTIT, Andrew T.**; PERKINS COIE LLP, P.O. Box 1247, Seattle, WA 98111-1247 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

THREAT DETECTION PLATFORMS FOR DETECTING, CHARACTERIZING, AND REMEDIATING EMAIL-BASED THREATS IN REAL TIME

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to US Application No. 16/672,854, titled "Threat Detection Platforms for Detecting, Characterizing, and Remediating Email-Based Threats in Real Time" and filed on November 4, 2019, which claims priority to US Provisional Application No. 62/782,158, titled "Systems and Methods for Email-Based Threat Detection" and filed on December 19, 2018, US Provisional Application No. 62/807,888, titled "Incident Detection and Management" and filed on February 20, 2019, and US Provisional Application No. 62/813,603, titled "Systems and Methods for Email-Based Threat Detection" and filed on March 4, 2019. Each of these applications is incorporated by reference herein its entirety.

TECHNICAL FIELD

[0002] Various embodiments concern computer programs and associated computer-implemented techniques for detecting email-based threats in the security field.

BACKGROUND

[0003] Employees of enterprise organizations (or simply "enterprises") will often receive malicious email in their inboxes. Some of these malicious emails are quite sophisticated. For instance, a malicious email that constitutes an attack on the employee or the enterprise may be designed to bypass existing protective measures, reach the employee's inbox, and then be opened. Such emails usually arrive unbeknownst to the security team of the enterprise.

[0004] Many employees will take action(s) upon receiving a malicious email that place data (e.g., her own personal data or enterprise data) at risk. For example, an employee may click on a link embedded in the malicious email, provide her credentials, send confidential information, or transmit money to the

unauthorized entity (also referred to as an "attacker" or "adversary") responsible for generating the malicious email. Performance of such actions may result in the installation of malicious software, theft of credentials, compromise of the employee's email account, exfiltration of data, or theft of money.

[0005] Upon discovering the breach, the enterprise faces significant ramifications. These ramifications include:

- Covering direct costs of the breach - especially if money was wired directly to the adversary;
- Covering indirect costs of the breach, such as infected hardware and labor to remediate the attack; and/or
- Covering fines assessed by regulatory agencies in the event of data theft.

[0006] Conventional email security software struggle to deal with attacks involving sophisticated malicious emails for several reasons.

[0007] First, an active adversary responsible for crafting personalized messages is often on the other side of sophisticated malicious emails. This is in contrast to less sophisticated email-based attacks, where a single person may send thousands or millions or generic, non-personalized emails in an attempt to succeed by sheer volume. Here, each sophisticated attack is new, unique, and personalized (e.g., for an employee or enterprise). Thus, employees will not observe the same sophisticated attack multiple times.

[0008] Second, sophisticated malicious emails generally will not include any attack signatures. The term "attack signature," as used herein, refers to a trait that has been previously observed in one or more emails determined to be malicious. Conventional solutions typically rely on attack signatures as well as pattern matching, but sophisticated malicious emails can obviate these conventional solutions through personalization. Moreover, some sophisticated malicious emails do not contain any links or attachments. Instead, a sophisticated malicious email may simply contain text, such as "Hey, can you handle a task for me?" Upon receiving a response, the adversary may instruct the employee to, for example, wire money or share data. Further, in the case of

an employee's email account, all emails will originate from the actual email account, thereby making it extremely difficult to detect malicious activity.

[0009] Third, the volume of emails handled by an enterprise is large, and receipt of emails is time sensitive. The decision on whether an email constitutes fraud should be rendered quickly for most emails as email security software should not inject delay into the flow of email. However, in most instances, conventional email security software indefinitely delays the delivery of email determined to represent a security threat.

[0010] Fourth, a relatively small number of sophisticated malicious emails will be handled by an enterprise over a given timeframe. For example, an enterprise may only observe several examples of sophisticated malicious emails over the course of a week. Accordingly, breaches due to sophisticated content are rare, and thus little data exists that can be ingested by machine learning (ML) models designed to identify sophisticated malicious emails.

[0011] Thus, there is a need in the security field to create computer programs and associated computer-implemented techniques for detecting email-based threats and then mitigating those threats.

[0011a] It is desired to address or ameliorate one or more disadvantages or limitations associated with the prior art, or to at least provide a useful alternative.

SUMMARY

[0011b] In at least one embodiment, the present invention provides a computer-implemented method comprising: establishing, via an application programming interface, a connection with a storage medium that includes information regarding digital conduct of employees of an enterprise, wherein the storage medium is managed by an entity that supports an office suite that is utilized by the employees of the enterprise; downloading, via the application programming interface, a first series of past communications received by an employee over a first interval of time into a local processing environment; building a machine learning (ML) model for the employee by providing the first

series of past communications to the ML model as training data, so as to train the ML model to understand normal traits and content of communications received by the employee; receiving, via the application programming interface, a communication addressed to the employee; and establishing whether the communication represents a security risk by applying the ML model to the communication.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] Various features of the technology will become more apparent to those skilled in the art from a study of the Detailed Description in conjunction with the drawings. Embodiments of the technology are illustrated by way of example and not limitation in the drawings, in which like references may indicate similar elements.

[0013] Figure 1 illustrates how a conventional filtering service can examine incoming emails to determine which emails, if any, should be prevented from reaching their intended destination.

[0014] Figure 2 illustrates how a threat detection platform may apply a multi-tiered ensemble model comprised of multiple sub-models to incoming emails received via the Internet to determine which emails, if any, should be prevented from reaching their intended destination.

[0015] Figure 3 depicts an example of a system for detecting email-based threats that includes a customer network (also referred to as an “enterprise network”) and a threat detection platform.

[0016] Figure 4 depicts a flow diagram of a process for detecting email-based threats by monitoring incoming emails, determining email attributes, detecting an attack based on the determined attributes, and optically performing remediation steps.

[0017] Figure 5 depicts an example of a hierarchical graph of possible attack types as generated by a machine learning (ML) model for a particular customer.

[0018] Figure 6 depicts an example of a threat detection platform that includes

multiple analysis modules and multiple extractors (e.g., multiple primary extractors and multiple secondary extractors) operating in conjunction with each other.

[0019] Figure 7 depicts how the vast majority of incoming messages may be classified as non-malicious while a small percentage of incoming messages may be classified as malicious.

[0020] Figure 8A includes a high-level illustration of the detection architecture of a threat detection platform in accordance with some embodiments.

[0021] Figure 8B includes a more detailed example of a process by which a threat detection platform can process data related to past emails (here, acquired from Microsoft Office® 365), extract primary attributes from the past emails, generate corpus statistics based on the primary attributes, derive secondary attributes based on the primary attributes and the corpus statistics, train ML model(s) with the primary attributes and/or the secondary attributes, and then employ the ML model(s) to score incoming emails based on the risk posed to an enterprise.

[0022] Figure 9 depicts an example of an incoming email that may be examined by the threat detection platform.

[0023] Figure 10A depicts how information gleaned from the incoming email can be used to establish different entities.

[0024] Figure 10B depicts an example of how the threat detection platform can execute the entity resolution procedure to establish the identity of the sender of the incoming email.

[0025] Figure 11 depicts how an entity risk graph can contain historical combinations of entities that were discovered in an incoming email and risk scores associated with those historical combinations.

[0026] Figure 12 depicts an example of an entity risk graph.

[0027] Figure 13 provides an example matrix of the stages that may be performed by a threat detection platform as it processes data, extracts features,

determines whether an event is representative of an attack, etc.

[0028] Figures 14A-H depict examples of different data structures that may be created/populated by the threat detection platform as it processes data, extracts features, determines whether an event is representative of an attack, etc.

[0029] Figure 15 includes a high-level system diagram of a threat intelligence system of which the threat detection platform is a part.

[0030] Figure 16 illustrates how a threat detection platform may derive/infer attributes from data acquired from various sources, profile those attributes to ML models as input, and then examine the output produced by those ML models to determine whether a security threat exists.

[0031] Figure 17 includes a high-level architectural depiction of a threat detection platform able to perform generate/update the data used for real-time processing of incoming emails via batch processing.

[0032] Figure 18A includes a high-level illustration of a process by which a threat detection platform can perform threat intelligence.

[0033] Figure 18B includes a high-level illustration of a process by which a threat detection platform can "productionalize" a signature to be used to determine the threat posed by incoming emails.

[0034] Figure 19A includes a high-level illustration of a process by which a threat detection platform can index corpus statistics to create a date-partitioned database of signatures and corpus statistics that can be used to identify unsafe entities more effectively.

[0035] Figure 19B depicts an example of a database that includes signatures and corpus statistics.

[0036] Figure 20 illustrates how the threat detection platform can detect instances of employee account compromise (EAC).

[0037] Figure 21 depicts a high-level flow diagram of a process for scoring the threat posed by an incoming email.

[0038] Figure 22 depicts a flow diagram of a process for applying a personalized machine learning (ML) model to emails received by an employee of an enterprise to detect security threats.

[0039] Figure 23 depicts a flow diagram of a process for detecting and characterizing email-based security threats in real time.

[0040] Figure 24 is a block diagram illustrating an example of a processing system in which at least some operations described herein can be implemented.

[0041] The drawings depict various embodiments for the purpose of illustration only. Those skilled in the art will recognize that alternative embodiments may be employed without departing from the principles of the technology. Accordingly, while specific embodiments are shown in the drawings, the technology is amenable to various modifications.

DETAILED DESCRIPTION

[0042] A significant portion of targeted attacks on enterprises or their employees start with email, and these security threats are always evolving. As discussed above, a significant need for detecting and then resolving sophisticated email-based threats is becoming increasingly apparent. Conventional email security software does not adequately address the need for accurate, fast, and consistent detection of sophisticated malicious emails before those emails enter an inbox.

[0043] While a variety of different attack types should be addressed by a solution, two particular attack types present challenges in terms of detection and resolution. The first attack type is email account compromise (also referred to "account takeover"). This form of attack is one in which an adversary accesses an employee's account with stolen credentials and then uses those credentials to steal money or data from the enterprise, or to send emails from the account in an effort to steal money or data from the enterprise or other employees. The second attack type is business email compromise. This form of attack is one in which an adversary impersonates an employee or a partner (e.g., a vendor). For

example, the adversary may cause incoming email to appear as though it has been written by an employee (e.g., by changing the display name). This form of attack is usually performed in an effort to have an invoice - either legitimate or fictional - paid by the enterprise or to steal data.

[0044] Introduced here are threat detection platforms designed to collect and examine emails in order to identify security threats to an enterprise. A threat detection platform (also referred to as an "email security platform") can be designed to address the above-mentioned attack types as well as other attack types such as phishing (e.g., campaign-based attacks), spear phishing (e.g., personalized attacks), extortion (e.g., cryptocurrency, gift card, and wire transfer ransoms), financial/data theft (e.g., vendor, partner, and client impersonation), and many other types of attacks, including those that have never been seen before.

[0045] At a high level, the technologies described herein can function to build a model representative of the normal email behavior of an enterprise (or an individual employee of the enterprise) and then look for deviations to identify abnormalities by applying the model to incoming emails. By establishing what constitutes normal behavior traits and/or normal email content, the enterprise can be protected against new, sophisticated attacks such as employee impersonation, vendor impersonation, fraudulent invoices, email account compromise, and account takeover. Moreover, canonicalizing, structuring, and storing data related to email may permit other high-value datasets to be created. For example, a threat detection platform may be able to derive valuable information regarding enterprise resource planning (ERP) from the email data. As further discussed below, the technologies described herein can leverage machine learning, heuristics, rules, human-in-the-loop feedback and labeling, or some other technique for detecting an attack (e.g., in real time or near real time) based on features extracted from a communication (e.g., an email) and/or context of the communication (e.g., recipient, sender, content, etc.).

[0046] Remediation action may be taken once a security threat has been detected. The remediation actions, if any, deemed appropriate may depend on the type of security threat that has been detected. For example, a threat detection platform may perform different remediation actions upon discovering a malicious email that includes an embedded link rather than a malicious email with an attachment. As part of the threat detection, identification, and remediation process, the threat detection platform may consider as input user actions; user-reported emails; machine learning (ML) training data including human-labeled emails, historical threat information, and scores; probabilities for threat detection based on models of known types of attacks; and heuristics including rules for blacklisting and/or whitelisting emails that meet certain criteria.

[0047] Introduced here are threat detection platforms designed to collect and examine emails in order to identify security threats to an enterprise. At a high level, the technologies described herein can function to build a model representative of the normal email behavior of an enterprise (or an individual employee of the enterprise) and then look for deviations to identify abnormalities by applying the model to incoming emails. By establishing what constitutes normal behavior traits and/or normal email content, the enterprise can be protected against new, sophisticated attacks such as employee impersonation, vendor impersonation, fraudulent invoices, email account compromise, and account takeover. As further discussed below, the technologies described herein can leverage machine learning, heuristics, rules, human-in-the-loop feedback and labeling, or some other technique for detecting an attack (e.g., in real time or near real time) based on features extracted from a communication (e.g., an email), attributes of the communication (e.g., recipient, sender, content, etc.), and/or datasets/information unrelated to communications. For instance, detecting sophisticated attacks that are plaguing an enterprise may require knowledge gleaned from multiple datasets. These datasets could include employee sign-in data, security events, calendar, contacts information, human resources (HR) information, etc. Each of these different datasets provides a

different dimension to the normality of behavior of an employee and can be used to detect the most sophisticated attacks.

[0048] Remediation action may be taken once a security threat has been detected. The remediation actions, if any, deemed appropriate may depend on the type of security threat that has been detected. For example, a threat detection platform may perform different remediation actions upon discovering a malicious email that includes an embedded link rather than a malicious email with an attachment.

[0049] Embodiments may be described with reference to particular network configurations, attack types, etc. However, those skilled in the art will recognize that these features are equally applicable to other network configurations, attack types, etc. For example, while certain embodiments may be described in the context of the spear phishing attacks, the relevant feature(s) may be used in conjunction with other types of attacks.

[0050] Moreover, the technology can be embodied using special-purpose hardware (e.g., circuitry), programmable circuitry appropriately programmed with software and/or firmware, or a combination of special-purpose hardware and programmable circuitry. Accordingly, embodiments may include a machine-readable medium having instructions that may be used to program a computing device to perform a process for receiving input indicative of an approval to access email messages that were delivered to, or sent by, employees of an enterprise over a given interval of time, establishing a connection with a storage medium that includes the email messages, downloading the email messages into a local processing environment, building a ML model for identifying abnormal communication behaviors based on characteristics of the email messages, etc.

Terminology

[0051] References in this description to “an embodiment” or “one embodiment” means that the particular feature, function, structure, or characteristic being described is included in at least one embodiment.

Occurrences of such phrases do not necessarily refer to the same embodiment, nor are they necessarily referring to alternative embodiments that are mutually exclusive of one another.

[0052] Unless the context clearly requires otherwise, the words "comprise" and "comprising" are to be construed in an inclusive sense rather than an exclusive or exhaustive sense (i.e., in the sense of "including but not limited to"). The terms "connected," "coupled," or any variant thereof is intended to include any connection or coupling between two or more elements, either direct or indirect. The coupling/connection can be physical, logical, or a combination thereof. For example, devices may be electrically or communicatively coupled to one another despite not sharing a physical connection.

[0053] The term "based on" is also to be construed in an inclusive sense rather than an exclusive or exhaustive sense. Thus, unless otherwise noted, the term "based on" is intended to mean "based at least in part on."

[0054] The term "module" refers broadly to software components, hardware components, and/or firmware components. Modules are typically functional components that can generate useful data or other output(s) based on specified input(s). A module may be self-contained. A computer program may include one or more modules. Thus, a computer program may include multiple modules responsible for completing different tasks or a single module responsible for completing all tasks.

[0055] When used in reference to a list of multiple items, the word "or" is intended to cover all of the following interpretations: any of the items in the list, all of the items in the list, and any combination of items in the list.

[0056] The sequences of steps performed in any of the processes described here are exemplary. However, unless contrary to physical possibility, the steps may be performed in various sequences and combinations. For example, steps could be added to, or removed from, the processes described here. Similarly,

steps could be replaced or reordered. Thus, descriptions of any processes are intended to be open-ended.

Technology Overview

[0057] Basic filtering services are offered by most email platforms. Figure 1 illustrates how a conventional filtering service can examine incoming emails to determine which emails, if any, should be prevented from reaching their intended destination. In some instances an enterprise applies an anti-spam filter to the incoming emails received via the Internet, while in other instances another entity, such as an email service, applies the anti-spam filter to the incoming emails on behalf of the enterprise. Emails received via the Internet 102 may be referred to as "external emails." The term "internal emails," meanwhile, may be used to refer to those emails that are sent within the enterprise. An example of an internal email is an intra-enterprise email (e.g., an email from one employee to another employee) that is delivered directly to the recipient mailbox rather than routed through the mail exchanger (MX) record, external gateways, etc.

[0058] Generally, the anti-spam filter 104 is designed to catch and quarantine malicious emails using blacklist(s) of senders, sender email addresses, and websites that have been detected in past unsolicited emails and/or policy frameworks defined by an enterprise. The term "anti-spam filter," as used herein can refer to any legacy email security mechanism capable of filtering incoming emails, including secure email gateways (SEGs) (also referred to as "gateways"). For example, the enterprise 108 (or the email service) may maintain a list of sender email addresses from which malicious email has been received in the past. As another example, an enterprise may decide to implement a policy that prohibits employees from receiving emails originating from a given domain. Malicious emails that are caught by the anti-spam filter 104 can be quarantined so as to remain hidden from the intended recipients, while non-malicious emails may be stored on an email server 106 (e.g., a cloud-based email server) for subsequent access by the intended recipients. Email servers (also referred to as

"mail servers") facilitate the delivery of emails from senders to recipients. Normally, an email will be transferred amongst a series of email servers as it travels toward its intended destination. This series of email servers allows emails to be send between dissimilar email address domains.

[0059] Email servers can be broken down into two main categories: outgoing mail servers and incoming mail servers. Outgoing mail servers may be referred to as Simple Mail Transfer Protocol (SMTP) servers. Incoming mail servers will generally be either Post Office Protocol Version 3 (POP3) servers or Internet Message Access Protocol (IMAP) servers. POP3 servers are known for storing sent/received messages on local hard drives, while IMAP servers are known for storing copies of messages on servers (though most POP3 servers can store messages on servers too). Thus, the location of emails received by an enterprise may depend on the type of incoming mail server used by the enterprise.

[0060] As discussed above, such an arrangement is not suitable for recognizing sophisticated malicious emails. Accordingly, conventional filtering services often allow sophisticated malicious emails to reach employees' inboxes by mistake. Introduced here, therefore, are threat detection platforms designed to improve upon conventional filtering services. Figure 2 illustrates how a threat detection platform 214 may apply a multi-tiered ensemble model comprised of multiple sub-models to incoming emails received via the Internet 202 to determine which emails, if any, should be prevented from reaching their intended destination.

[0061] Initially, the threat detection platform 214 may receive an email addressed to an employee of an enterprise. Upon receiving the email, the threat detection platform 214 may apply a first model 204 to the email to produce a first output indicative of whether the email is representative of a malicious email. The first model may be trained using past emails addressed to employee(s) of the enterprise that have been verified as non-malicious emails. Accordingly, the first

model 204 may be referred to as the "surely-safe model." The first model 204 serves as the first level of threat detection, and therefore may be tuned/designed to permit most email (e.g., upwards of 90, 95, or 99 percent of all incoming email) to reach the email server 206. Normally, the first model 204 is designed such that the initial threat determination is made fairly quickly (e.g., in less than 100, 50, or 25 milliseconds). Thus, the first model 204 may be responsible for performing load shedding.

[0062] If the email cannot be verified as non-malicious by the first model 204, the threat detection platform 214 can apply a second model 208 to the email. For the purpose of illustration, emails forwarded to the second model 204 may be referred to as "malicious emails." However, these emails can be more accurately described as possibly malicious emails since the first model 204 is only able to verify whether emails are non-malicious. Upon being applied to the email, the second model 208 may produce a second output indicative of whether the email is representative of a given type of malicious email. Generally, the second model 208 is part of an ensemble of models applied to the email responsive to determining that the email is representative of a malicious email. Each model in the ensemble may be associated with a different type of security threat. For example, the ensemble may include separate models for determining whether the email includes a query for data/funds, a link to a Hypertext Markup Language (HTML) resource, an attachment, etc. As further discussed below, the second model 208 may be designed to establish different facets of the security threat responsive to a determination that the email is likely malicious. For instance, the second model 208 may discover facets of the security threat such as the strategy, goal, impersonated party, vector, and attacked party, and then upload this information to a profile associated with the intended recipient and/or the enterprise.

[0063] Then, the threat detection platform 214 may apply a third model 210 designed to convert the output produced by the second model 208 into a comprehensible visualization component 212. In embodiments where the

second model 208 is part of an ensemble of models, the third model 210 can aggregate the outputs produced by the models in the ensemble, characterize the attack based on the aggregated outputs, and then convert the aggregated outputs into an explainable insight. For example, the third model 210 may generate a notification that identifies the type of security threat posed by the email, whether remediation actions are necessary, etc. As another example, the third model 210 may generate a human-readable insight (e.g., that includes text, graphics, or some combination thereof) using the facets, model features, and/or most discriminating features that triggered the determination that a security threat exists for the combination of email and attack. The explainable insight may be created so that security professionals responsible for addressing/mitigating security threats can more easily understand why the second model 208 flagged an incoming email as being representative of an attack.

[0064] Those skilled in the art will appreciate that the output of one model may be the entrance criteria for another model. Said another way, the order of the models employed by the threat detection platform 214 may triage which emails go to which models in an effort to reduce analysis time. Thus, the threat detection platform 214 may take a hierarchical bi-phasal approach to examining incoming emails.

[0065] The multi-tiered ensemble model may be comprised of different types of models, such as gradient boosting decision tree (GBDT) models, logistic regression models, and/or deep learning models. As further discussed below, each type of attack is generally scored by a separate model, so the threat detection platform 214 may employ different types of models based on the type of attack being detected.

[0066] Such an approach may be referred to as a "bi-phasal approach," as it allows emails determined to be non-malicious to be routed to the email server 206 with very little delay while additional time is spent analyzing emails determined to be malicious (or at least potentially malicious).

Threat Detection Platform for Detecting Email-Based Threats

[0067] Figure 3 depicts an example of a system 300 for detecting email-based threats that includes a customer network 316 (also referred to as an "enterprise network") and a threat detection platform 302. As shown in Figure 3, the threat detection platform 302 may include a profile generator 304, a training module 306, a monitoring module 308, a threat detection datastore 310, an analysis module 312, and a remediation 314. Some embodiments of the threat detection platform 302 include a subset of these components, while other embodiments of the threat detection platform 302 include additional components that are not shown in Figure 3.

[0068] The system 300 can function to acquire email usage data of a customer (also referred to as an "enterprise"), generate a profile based on the email usage data that includes a number of received or inferred behavioral traits, monitor incoming emails, and, for each email, determine whether the email represents a security threat using a set of attack detectors (e.g., based on a deviation from behavioral trait(s) or normal content, such as by feeding the deviations into ML models), flag the email as a possible attack if the detectors indicate one, and if flagged, optionally perform one or more remediation steps on the email. The remediation step(s) may be performed in accordance with a customer-specified remediation policy and/or a default remediation policy. The term "customer," as used herein, may refer to an organization (e.g., a corporation or an enterprise), a business unit, an individual (e.g., associated with one or more email addresses, a team, or any other suitable set of users of the threat detection platform 302. While embodiments may be described in the context of enterprises, those skilled in the art will recognize that the relevant technologies may be applied to other types of customers. As further discussed below, the system 300 may train one or more ML modules to serve as detectors capable of detecting a plurality of email attack types that can appear in the incoming email based on deviations from customer behavioral traits, normal email content, etc.

[0069] In some embodiments, the system 300 detects attacks based on the entire email (e.g., including the content of the body). However, in other embodiments, the system 300 is designed to detect attacks based only on the email metadata (e.g., information regarding the email header, sender, etc.) or some other suitable data.

[0070] All or portions of the system 300 can be implemented in an entity's email environment (e.g., the customer network 316), a remote computing system (e.g., through which incoming emails and/or data regarding incoming emails can be routed through for analysis), the entity's gateway, or another suitable location. The remote computing system can belong to, or be maintained by, the entity, a third-party system, or another suitable user. The system 300 may be integrated into the entity's email system: inline (e.g., at the secure email gateway), via an application programming interface (API) (e.g., wherein the system receives email data via an API such as the Microsoft Outlook® API), or another suitable manner. Thus, the system 300 can supplement and/or supplant other communication security systems employed by the entity.

[0071] In a first variation, the system 300 is maintained by a third party (also referred to as a "threat detection service") that has access to multiple entities' emails. In this variation, the system 300 can route the emails, extracted features (e.g., primary attribute values), derivatory information (e.g., secondary attribute values), and/or other suitable communication information to a remote computing system maintained/managed by the third party. The remote computing system may be, for example, an instance on Amazon Web Services (AWS). In this variation, the system 300 can maintain one or more databases for each entity that includes, for example, organizational charts, attribute baselines, etc. Additionally or alternatively, the system 300 may maintain federated database(s) shared amongst multiple entities such as detector databases, legitimate vendor databases, etc. In this variation, the third party can maintain different instances of the system 300 for different entities, or maintain a single instance for multiple entities. The data hosted in these instances can be obfuscated, encrypted,

hashed, de-personalized (e.g., by removing personal identifying information (PII)), or otherwise secured or secreted.

[0072] In a second variation, the system is maintained by the entity whose emails are being monitored (e.g., remotely or on premises), and all data can be hosted by the entity's computing system. In this variation, data to be shared across multiple entities, such as detector database updates and new attack signatures, can be shared with a remote computing system maintained by a third party. This data can be obfuscated, encrypted, hashed, de-personalized (e.g., by removing PII), or otherwise secured or secreted. However, the system 300 can be maintained or executed using any other suitable computing and ownership configuration.

[0073] As shown in Figure 3, the profile generator 304, training module(s) 306, monitoring module(s) 308, threat detection datastore 310, analysis module(s) 312, and remediation engine(s) 314 can be part of a threat detection platform 302. Alternatively, these components can be individually used and/or implemented. The threat detection platform 302 may be implemented by a threat detection service (also referred to as a "computer security service"), a customer (e.g., an enterprise, organization, or individual that has an account or is otherwise implementing threat detection services), an entity/individual associated with (or representative of) a customer, a trusted third party, or any other service, entity, or individual. In some embodiments, one or more aspects of the system 300 may be enabled by a web-accessible computer program operable on a computer server or a distributed computing system. For example, an individual may be able to interface with the threat detection platform 302 through a web browser executing on a computing device.

[0074] The customer network 316 can be an enterprise network, mobile network, wired network, wireless network, wireless spectrum network, or any other communications network maintained by a customer or a network operator associated with the customer. As noted above, the customer can be an

individual, an enterprise, or another suitable entity. For example, an enterprise may utilize the services of a computer security company for at least email threat detection. The enterprise may grant permission to the computer security company to monitor the customer network 316, including monitoring incoming emails at the customer network 316, analyzing the emails for potential threats, and performing some remediation action(s) if threats are detected. In some embodiments, the enterprise further grants permission to the computer security company to collect or receive various pieces of data about the enterprise in order to build a profile that specifies enterprise norms, behavioral traits, and normal email content.

[0075] The threat detection datastore 310 may include one or more databases in which customer data, threat analysis data, remediation policy information, customer behavioral traits or norms, normal customer email content, and other pieces of data can be stored. The data can be: determined by the system 300 (e.g., calculated or learned from data retrieved, received, or otherwise collected from the customer network 316 or the entity's email provider), received from a user, retrieved from an external database (e.g., LinkedIn® or Microsoft Office 365®), or otherwise determined. In some embodiments, the threat detection database 310 also stores output from the threat detection platform 302, including human-readable information on detected threats and actions taken. Other various data or entities may be stored.

[0076] Customer data can include, but is not limited to, email usage data; organizational data such as organization members/employees and their titles; customer behavior traits or norms (e.g., determined based on historical emails); attack history (e.g., determined based on historical emails, determined by applying the attribute extractors and/or analysis modules to historical emails, etc.); entity profile(s); normal customer email content; email addresses and/or phone numbers of organization members; identities of entities and/or individuals who organization members frequently communicate with, both internally and

externally; email volume during various times of the day; topics or subject matter most frequently discussed, and how frequently; and more.

[0077] The system 300 can optionally include a profile generator 304 that generates one or more entity profiles (also referred to as "customer profiles") based on past emails and/or email usage data associated with the entity. In a second variation, the system 300 includes multiple profile generators 304 that each extract one or more attributes of the entity profile. However, the system 300 can include any suitable number of profile generators in any suitable configuration.

[0078] Entity profiles can be generated for: each customer, each business unit, each individual (e.g., each employee or recipient of emails), each email address, each organization type, or another suitable entity or entity group. The entity profile is preferably used as the baseline for entity communication behavior (e.g., email behavior) but could be used in other manners. Moreover, profiles could be generated external to an entity, and these profiles may be federated across a customer base for use by all entities whose emails are being monitored by the system 300. For example, a profile for a trusted third party (e.g., Oracle), a representative of a trusted third party (e.g., a sales representative at Oracle), or a financial institution (e.g., with known routing numbers so as to detect fraudulent invoice payments) could be federated across a customer base. Thus, the system 300 may build a federated network of profiles modeling businesses, vendors, customers, or people.

[0079] An entity profile can include: primary attributes, secondary attributes, or any other suitable feature. These values can be: median values, mean values, standard deviations, ranges, thresholds, or any other suitable set of values (e.g., for the entity profile, extracted from new emails, etc.). The entity profile can include: time series (e.g., trends or values for a specific recurrent time, such as months of the year), static values, or may have other suitable contextual dependency.

[0080] Primary attributes are preferably attributes or features extracted directly from a communication, but could be otherwise determined. The primary attributes can be extracted by one or more primary attribute extractors, each extracting one or more primary attributes from the communication as shown in Figure 6, but could be otherwise extracted. The primary attribute extractor(s) can be global (e.g., shared across multiple entities), specific to an entity, or otherwise shared. Examples of primary attributes include the sender's display name, sender's username, Sender Policy Framework (SPF) status, DomainKeys Identified Mail (DKIM) status, number of attachments, number of links in the email body, spam or phishing metrics (e.g., continent or country of origin), whether data between two fields that should match are mismatched, header information, or any other suitable communication data. Primary attributes can optionally include metadata attributes (e.g., company identifier (ID), message ID, conversation ID, individual ID, etc.).

[0081] Secondary attributes are preferably attributes that are determined from the primary attributes and/or customer data (e.g., as determined from the threat detection datastore 310), but can be otherwise determined. The secondary attributes can be extracted, inferred, calculated, or otherwise determined. The secondary attributes may be determined by one or more secondary attribute extractors, each extracting one or more secondary attributes from the primary attributes for a given communication or entity as shown in Figure 5, but could be otherwise determined. The secondary attribute extractors can be global (e.g., shared across multiple entities), specific to an entity, or otherwise shared. The secondary attributes can be determined from a time series of primary attribute values (e.g., wherein each primary attribute value can be associated with a timestamp, such as the sent timestamp or receipt timestamp of the email), from all primary attribute values, from a single primary attribute value, from the values of multiple primary attributes, or from any other suitable set of data. Examples of secondary attributes can include: frequencies, such as sender frequencies (e.g., sender fully qualified domain name (FQDN) frequencies, sender email frequencies, etc.) or domain frequencies (e.g., SPF status frequencies for a given

domain, DKIM status frequencies for a given domain, the frequency at which the system receives the same or similar email body from a given domain, how frequently emails are received from that domain, how frequently emails are sent to that domain, etc.); determining a mismatch between one or more primary attributes that should match; employee attributes (e.g., name, title, whether the entity is employed, whether the entity has a high attack risk, whether the entity is suspicious, whether the entity has been attacked before, etc.); vendor attributes (e.g., vendor name, whether the vendor is an exact match with a known vendor, whether there is a vendor Unicode lookalike, etc.); whether the body of the communication includes one of a set of high-risk words, phrases, sentiments, or other content (e.g., whether the communication includes financial vocabulary, credential theft vocabulary, engagement vocabulary, non-ASCII content, attachments, links, etc.); domain information (e.g., domain age, whether the domain is blacklisted or whitelisted, whether the domain is internal or external, etc.); heuristics (e.g., whether the FQDN, domain, domain name, etc., has been seen before, either globally or by the entity); primary attribute value (e.g., as extracted from the communication) deviation from the respective baseline value (e.g., deviation magnitude, whether the value has deviated beyond a predetermined variance or difference threshold); or any other suitable attribute, feature, or variable. In some embodiments, the secondary attributes are determined as a function of the primary attributes. One example of a primary attribute is a sender email address, while one example of a secondary attribute is the statistics of communications patterns from sender address to recipient, department, organization, and universe of customers.

[0082] The entity profile can additionally or alternatively include: a number of customer behavioral traits or typical email content associated with the customer. In some embodiments, the profile generator 304 receives the email usage data from the customer network 316 or the threat detection datastore 310. The email usage data can include, but is not limited to, information on email addresses of employees and contacts, email content (e.g., the body of email messages), frequency of email, volume of email at given times of day, HTML/font/style usage

within email, confidential topics and explicitly or implicitly authorized members discussing those topics, spam mail and characteristics thereof, and more.

[0083] The entity profile can be generated from: historic email data for the entity (e.g., retrieved using an API to the entity's email environment, retrieved from an email datastore, etc.); newly-received email (e.g., email received after system connection to the entity's email environment); user inputs; other entities (e.g., sharing a common feature or characteristic with the entity); or based on any other suitable data. In some embodiments, one or more pieces of email usage data can be collected, generated, or inferred by the profile generator 304 based on received pieces of customer data, monitoring of the customer network 316 given authentication and access by the customer, or some other means.

[0084] The entity profile can be generated using the same system(s) as typical email analysis (e.g., using the attribute extractors used to extract the attributes for real- or near-real-time threat detection), but can alternatively or additionally be generated using other suitable system(s).

[0085] In one variation, the profile generator 304 generates a customer profile by building a deep profile of the company's email usage, membership roles and/or hierarchy, daily norms, behavioral traits, and more in order to establish a model of what "normal" or "typical" looks like for the customer in terms of email usage and behavior, and, by extension and inference, what "abnormal" or "atypical" emails and/or activity may constitute for purposes of identifying likely threats.

[0086] In some embodiments, the customer profile is generated based on received, collected, and/or inferred customer data, email usage data, and other relevant information. Examples of questions that the customer profile may seek to model answers for regarding the customer include, but are not limited to: What are normal email addresses for each member of the organization? What are normal topics for each person, pair, and/or department to be discussing (e.g., Joe and Samantha normally discuss product release plans, but never accounting

or billing topics)? What are normal login or email sending times for each user? What Internet Protocol (IP) address do they log in from? What geolocation do users usually log in from? Do users have suspicious mail filter rules set up (e.g., attackers who hijack email accounts sometimes set up mail filters to auto-delete emails containing certain keywords in order to conceal illicit activity from the account's true owner)? What's the normal tone or style each user uses? What's the tone used between each pair of users? What's the normal signature (e.g., "cheers" or "thanks") used by each employee? What types of words are used more in one department and less in another department? What are the normal vendors/partners that a customer communicates with and/or is billed by? Does a given pair of users usually talk? What are typical email authentication statuses for this person, pair, or entity (e.g., SPF, DKIM, or Domain-Based Message Authentication, Reporting, and Conformance (DMARC))? When a user receives or sends links/attachments, does a derived description of the link/attachment match a given description of the link/attachment? When an employee receives emails with attachments, what are the typical characteristics of the attachments (e.g., name, extension, type, size)?

[0087] The monitoring module 308 operates to monitor incoming emails at a network maintained by the customer. In some embodiments, the monitoring module 308 monitors incoming emails in real time or substantially real time. In some embodiments, the monitoring module 308 is authorized to monitor the incoming emails only upon the system 300 and/or the threat detection platform 302 being authenticated and granted permission and access by the customer network 316. In some embodiments, the system 300 and/or the threat detection platform 302 is integrated into an office suite or an email suite via an API.

[0088] The analysis module 312 operates to analyze each incoming emails for possible threats/attacks. The analysis module 312 preferably detects attacks based on the secondary attributes (e.g., for one or more communications for the entity), but can alternatively or additionally detect attacks based on the primary attributes or any other suitable data. In one variation, the analysis module 312 is

separate from the primary and secondary attribute extractor(s). In another variation, the analysis module 312 can include the primary and/or secondary attribute extractor(s). However, the analysis module 312 can be otherwise configured.

[0089] The system 300 can include one or more analysis modules 312, operating in parallel, in series, or in another suitable order. An example of multiple analysis modules 312 operating in conjunction with each other is shown in Figure 6. The set of analysis modules 312 used for a given entity or communication can be: predetermined, manually determined, selected based on historical communications, selected based on operation context (e.g., fiscal quarter), or otherwise determined. In a first variation, the system 300 includes one or more analysis modules 312, of the same type or different types, for each known attack type. For example, each attack type may be associated with a different analysis module 312. In a second variation, the system 300 includes a single analysis module 312 for all attack types. In a third variation, the system 300 includes one or more analysis module for each attack type (e.g., a first set for phishing attacks, a second set for impersonation attacks, etc.). In a fourth variation, the system 300 includes a cascade/tree of analysis modules 312, where a first layer of analysis module(s) classifies incoming email with a potential attack type and subsequent layer(s) of analysis module(s) analyze whether the email has the characteristics of the attack type. Figure 5 depicts an example of a hierarchical graph of possible attack types as generated by a ML model for a particular customer, as described above with respect to the training module 306. In this example, high-level classifications include impersonation technique, attack vector, impersonated party, attacked party, and attack goal. Within the impersonation technique classification, attack types may include spoofing a user's name, spoofing a user's email, compromising an account, or none due to unknown sender. Based on the attack goals, attack types may include payroll fraud, stealing credentials, encouraging a user to wire money, bitcoin ransom, wire money ransom, etc.

[0090] However, the system 300 can include any suitable number of analysis modules 312 for detecting any number of attack types. In particular, the approach of modeling employed by the system 300 to discover behavioral norms of employee, vendor, and organizational communication allows for the ability to identify attacks that have not been seen before, as well as zero-day phishing attacks.

[0091] The analysis modules 312 can include or use one or more of: heuristics, neural networks, rules, decision trees (e.g., gradient-boosted decision trees), ML-trained algorithms (e.g., decision trees, logistic regression, linear regression, etc.), or any other suitable analysis method/algorithm. The analysis modules 312 can output: discrete or continuous outputs, such as a probability (e.g., attack probability), a binary output (e.g., attack/not attack), an attack classification (e.g., classification as one of a plurality of possible attacks), or output any other suitable output. The analysis modules 312 can be: received from a database (e.g., a database of known attack patterns or fingerprints), received from a user, learned (e.g., based on data shared across multiple entities, based on communication data for the entity, etc.), or otherwise determined.

[0092] Each analysis module can be specific to an attack, an attack type/class, or any other suitable set of attacks. The system 300 can include one or more analysis modules 312 per attack set. In one variation, the attack set can be associated with multiple analysis modules 312, where the system 300 can dynamically select the analysis module to use (and/or which output to use) based on the performance metrics of each analysis module for the given attack set. For example, the system 300 can include a heuristics-based analysis module and a ML-based analysis module for a given attack that are executed in parallel for each piece of communication; monitor the recall and/or the precision (e.g., as determined based on entity feedback of the email classifications) for both analysis modules; and select the analysis module with the higher performance value for subsequent communication analysis. The output of all but the highest-

performant analysis module can be hidden from the user and/or not used in email attack classification; alternatively, the outputs of lower-performant analysis modules can be used to verify the highest-performant analysis module output, or otherwise used.

[0093] One or more of the analysis modules 312 can be specific to an entity (e.g., an organization, a business organization, a title, an individual, an email address, etc.), be shared between multiple entities (e.g., be a global analysis module), or be otherwise customized or generic.

[0094] In one example, first, for each incoming email, the analysis module 312 (e.g., secondary attribute extractor) determines a deviation of the mail from each of the plurality of customer behavioral traits or content norms. In some embodiments, the deviation is a numerical value or percentage representing a delta between the customer behavioral trait and an assigned behavioral trait determined from the incoming email. For example, if a customer behavioral trait is "Joe Smith almost exclusively sends email from js@customerentity.com" and an incoming email purporting to be from Joe Smith has the email address joesmith@genericmail.com, then the deviation will be assigned a high number. If Joe Smith sends from a genericmail.com account approximately 20 percent of the time, then the deviation will still be relatively high, but the deviation will be lower than the previous example. Second, the analysis module 312 feeds the measured deviations as input into one or more attack detectors (e.g., rules-based engines, heuristics engines, ML models, etc.), each of which can generate an output. Third, the analysis module 312 flags the email as a possible attack corresponding to an email attack type if an indication is received from one or more of the ML models that a deviation threshold for the email attack type has been exceeded. The analysis module 312 can instruct the ML models to categorize the deviations in an incoming email as indicating a likely malicious email or a likely non-malicious email, as well as categorize the email according to a possible attack type. In some embodiments, the ML model "trips" - that is,

f(email) exceeds a threshold for deviations from customer behavioral traits and content norms - and then flags the email as a possible attack.

[0095] The output(s) produced by the analysis module(s) 312 can optionally be fed into a master detector that analyzes these output(s) in order to produce a final classification for the communication as an attack or non-attack, as shown in Figure 6. The master detector can optionally output the factors, rules, weights, variables, decision tree nodes, or other attack detector parameters that contributed to the attack classification.

[0096] The remediation engine 314 optionally operates to perform one or more remediation processes. The remediation engine 314 is preferably implemented in response to communication classification as an attack (e.g., by one or more analysis modules 312, by the master detector, etc.), but can alternatively or additionally be implemented at any other suitable time. In some embodiments, the remediation steps are based on or correlate to a customer remediation policy. The customer remediation policy may be predefined and received by the threat detection platform 302, be generated based on inference, analysis, and customer data, or be otherwise determined. In some embodiments, the threat detection platform 302 may prompt the customer to provide one or more remediation steps or components of the remediation policy in various situations. Remediation steps may include, for example, moving the email to a junk folder as shown in Figure 6, moving the email to a hidden folder, permanently deleting the email, performing different measures according to how users have acted on it, sending notifications to users (e.g., employees, administrators, members of a security team), resetting the password of the affected employee, ending all sessions, pushing signatures to a firewall or an endpoint protection system, pushing signatures to an endpoint protection system to lock one or more computing devices, etc. For example, upon discovering a compromised account, the threat detection platform 302 may invoke API(s) to block the compromised account, reset connections with services/databases, or change the password through a workflow. Additionally or alternatively,

remediation steps can include moving the email from the junk folder back into the inbox (e.g., in response to determination that the email was not an attack).

[0097] In some embodiments, the remediation engine 314 provides threat detection results and/or other output to the customer via, for example, a customer device 318. Examples of customer devices 318 include mobile phones, laptop computers, and other computing devices. In some embodiments, the remediation engine 314 sends the output in a human-readable format to the threat detection platform 302 for display on an interface.

[0098] The system 300 can optionally include a training module 306 that operates to train the ML model(s) employed by the analysis module(s) 312. Each ML model preferably detects a single attack type, but can alternatively detect multiple attack types. In some embodiments, the training module 306 trains the ML models by feeding training data into the ML models. The training data can include: entity-labeled communications (e.g., system-analyzed emails that have been sent to security personnel and labeled as attacks or non-attacks, as shown in Figure 6), third-party-labeled communications, or any other suitable set of communications. In some embodiments, the customer data, ML models, and/or thresholds are all different per customer as a result of the unique customer's behavioral traits being fed into the ML models to generate custom analyses. In some embodiments, the training data ingested by the models includes labeled datasets of "bad" emails, which are received or generated by one or more components of the threat detection platform 302. In some embodiments, the labeled datasets of bad emails include human-labeled emails. Through human labeling from, for example, customer administrators, network operators, employees, or security service representatives, a solid body of malicious emails can be built up for a customer, and these malicious emails can be used to train the ML models based on that customer. In some embodiments, the training data includes the received, collected, or inferred customer data and/or email usage data. In some embodiments, the training data can include historical threats that have been previously identified in customer inboxes. In

some embodiments, different ML models employed have been developed for different known types of attacks. In some embodiments, emails are scored, weighted, or assigned a percentage or numerical value based on using these ML model(s). In some embodiments, if an email scores over the threshold for any of the ML models, it may be flagged, unless a heuristic or other element of the threat detection platform 302 indicates that it should not be flagged.

[0099] In some embodiments, the training data for training the ML models can include human inputs received from the customer. Organizations often have phishing mailboxes where employees can report emails to security teams, or where security teams can automatically/manually reroute messages meeting certain criteria. The training data can include emails that are placed in these phishing mailboxes as malicious emails. In some embodiments, the human inputs can include end user actions that can feed into ML models. For example, if an individual moves an email that the ML models cannot decide with certainty whether to discard, that user action can be included as training data to train the ML models as to what action should be taken in similar contexts.

[00100] Examples of potential attack types that the ML models can be trained to detect in varying embodiments include, but are not limited to, vendor impersonation and ransom attacks.

[00101] In some embodiments, a plurality of heuristics data is utilized as an alternative to, or in combination with, the ML models to detect threats, train the ML models, infer behavioral traits or content norms for a customer based on customer data, select potential attack types of a customer, or perform other threat detection tasks. In some embodiments, one or more aspects of training the ML models includes feeding the plurality of heuristics data as input training data into one or more of the ML models. In some embodiments, the heuristics data is utilized in relation to a rules engine which operates to decide which heuristics to apply in different situations. In some embodiments, the rules engine determines whether to apply machine learning or heuristics in a particular threat

detection task. In some embodiments, one or more rules may include a blacklist and/or a whitelist for certain email criteria.

[00102] In some embodiments, any level of granularity can be contemplated for the system 300 regarding the analysis module 312, ML models, heuristics, rules, and/or human labeling of inputs. In some embodiments, "normal" and "abnormal" behavioral traits and content norms could be determined on a per-employee, per-pair, per-department, per-company, and/or per-industry basis.

[00103] In some embodiments, the ML models can optically be refined in a number of ways in the course of operation. In some embodiments, the monitoring module 308 monitors the customer's phishing mailbox to locate false negatives (i.e., emails that were missed by the ML models that employees subsequently reported to the security team). In some embodiments, a customer may reverse the remediation decision made by the heuristics and/or the ML models, and in response the ML models may incorporate that feedback. In some embodiments, if a customer marks a particular feature in an email (e.g., sender mail, display name, authentication state, etc.) as suspicious, that can feed back into the ML models. In some embodiments, such feedback is weighted in a ML model based on the stature or reputation of the individual responsible for providing the feedback. For example, a ML model may trust a Tier 3 employee's judgment on an email significantly more than a Tier 1 employee's judgment, and would weigh their feedback more heavily into the ML model.

[00104] In some embodiments, different types of ML models may be used, including but not limited to gradient-boosted decision trees, logistic regression, linear regression, and more. In some embodiments, ML models are replaced with purely rules-based engines.

[00105] Figure 4 depicts a flow diagram of a process 400 for detecting email-based threats by monitoring incoming emails (step 404), determining email attributes (step 405), detecting an attack based on the determined attributes (step 406), and optically performing remediation steps (step 408). In one

example, the process 400 can include collecting email usage data (step 401), generating an entity profile based on the email usage data (step 402), monitoring incoming emails (step 404), determining deviations in the incoming email, feeding the measured deviations into the ML models, flagging the email as a possible attack (step 407), and performing remediation steps (step 408). The process 400 can optionally include training ML models to detect email attack types (step 403).

[00106] The process 400 functions to provide email-based threat detection according to a generated customer profile that models normal customer behavior and normal email content, and then feeds the deviations from these normal behavior traits and normal content as input into ML models training on malicious emails.

[00107] In some embodiments, the process 400 is enabled by a web-based platform (e.g., threat detection platform 302 of Figure 3) operable on a computer server or a distributed computing system. Additionally or alternatively, the process 400 may be performed on any suitable computing device capable of ingesting, processing, and/or analyzing customer data and email usage data, performing ML techniques, and/or performing remediation actions.

[00108] The process 400 can be performed in parallel or in series with email delivery to an email inbox. In one variation, the process 400 is performed in parallel with email delivery to the recipient's email inbox, wherein the email is retroactively removed from the email inbox in response to determination that the email is an attack (and/or has a high probability of being an attack). In a second variation, the process 400 is performed in series with email delivery, where the email is only delivered to the recipient's email inbox in response to determination that the email is not an attack. However, the process 400 can be otherwise integrated into email delivery paradigms. The method can analyze: all incoming emails, only emails labeled as non-attacks by a preceding security system, only

emails labeled as attacks by a preceding security system, or any suitable set of emails.

[00109] Monitoring incoming emails (step 404) is preferably performed using a monitoring module (e.g., monitoring module 308 of Figure 3), as discussed above, but emails can be otherwise ingested.

[00110] Email attributes are preferably determined by extractors, as discussed above, but can be otherwise determined. In one example, the method includes: extracting primary attributes from the incoming email (e.g., with one or more specialized primary attribute extractors executing in parallel), and determining secondary attributes for the email (e.g., with one or more specialized secondary attribute extractors executing in parallel) from the primary attributes and customer data.

[00111] The attack is preferably determined using one or more analysis modules, as discussed above, but can be otherwise determined. In one variation, the determined attributes (e.g., primary or secondary attributes) can be fed into one or more analysis modules (e.g., executing in parallel or in series). In some embodiments, each analysis module is specific to an attack type, where the plurality of outputs from the plurality of analysis modules is further analyzed (e.g., by a master detector) to determine whether the email is an attack. In other embodiments, the analysis module detects multiple attack types (e.g., outputs multiple output values, each corresponding to a different attack type, where the output can be a likelihood and/or confidence in the corresponding attack type), and the email can be labeled as an attack when the output value exceeds a predetermined threshold for the corresponding attack type. However, the attack can be otherwise detected.

[00112] Step 408 optically includes performing remediation steps, as described above with respect to remediation engine 314 of Figure 3, but the email can be otherwise remediated.

[00113] Step 401 includes collecting or receiving email usage data, as described above with respect to profile generator 304 of Figure 3.

[00114] Step 402 includes generating a customer profile based on the email usage data, as described above with respect to profile generator 304 of Figure 3.

[00115] Step 403 includes training ML models to detect email attack types, as described above with respect to training module 306 of Figure 3.

[00116] Step 405 includes measuring deviations in incoming email, as described above with respect to analysis module 312 of Figure 3.

[00117] Step 406 includes feeding the measured deviations into ML models, as described above with respect to analysis module 312 of Figure 3.

[00118] Step 407 optically includes flagging email as a possible attack, as described above with respect to analysis module 312 of Figure 3.

Integrative Approach to Detecting Security Threats

[00119] As discussed above, conventional email filtering services are not suitable for recognizing sophisticated malicious emails, and therefore may allow sophisticated malicious emails to reach employees' inboxes by mistake. Introduced here are threat detection platforms designed to take an integrative approach to detecting the security threats to an enterprise.

[00120] Unlike conventional email filtering services, a threat detection platform can be completely integrated within the enterprise environment. For example, the threat detection platform may receive input indicative of an approval by an individual (e.g., an administrator associated with the enterprise or an administrator of the email service employed by the enterprise) to access email, active directory, mail groups, identity security events, risk events, documents, etc. The approval may be given through an interface generated by the threat detection platform. For example, the individual may access the interface

generated by the threat detection platform and then approve access to these resources as part of a registration process.

[00121] Upon receiving the input, the threat detection platform can establish a connection with storage medium(s) that include these resources via application programming interface(s) (APIs). For example, the threat detection platform may establish, via an API, a connection with a computer server managed by the enterprise or some other entity on behalf of the enterprise.

[00122] The threat detection platform can then download resources from the storage medium(s) to build a ML model that can be used to identify email-based security threats. Thus, the threat detection platform can build a ML model based on retrospective information in order to better identify security threats in real time as emails are received. For example, the threat detection platform may ingest incoming emails and/or outgoing emails corresponding to the last six months, and then the threat detection platform may build a ML model that understands the norms of communication with internal contacts (e.g., other employees) and/or external contacts (e.g., vendors) for the enterprise. Thus, actual threats rather than synthetic threats can be used to identify whether an incoming email is representative of a security threat.

[00123] Such an approach allows the threat detection platform to employ an effective ML model nearly immediately upon receiving approval from the enterprise to deploy it. Most standard integration solutions, such as anti-spam filters, will only have access going forward in time (i.e., after receiving the approval). Here, however, the threat detection platform may employ a backward-looking approach to develop personalized ML model(s) that are effective immediately. Moreover, such an approach enables the threat detection platform to go through a repository of past emails to identify security threats residing in employees' inboxes.

[00124] The aforementioned API-based approach provides a consistent, standard way of looking at all email handled by an enterprise (or another entity,

such as an email service, on behalf of the enterprise). This includes internal-to-internal email that is invisible from standard integration solutions. A SEG integration, for example, that occurs through the mail exchanger (MX) record will only be able to see incoming email arriving from an external source. The only way to make email arriving from an internal source visible to the SEG integration would be to externally reroute the email through the gateway.

[00125] The threat detection platform can design the ML model to sort emails determined to be possible security threats into multiple categories. Figure 7 depicts how the vast majority of incoming messages may be classified as non-malicious while a small percentage of incoming messages may be classified as malicious. Here, for example, nearly 99.99% of incoming messages have been classified as non-malicious, and thus immediately forwarded to the appropriate inbox. However, the threat detection platform has discovered three types of security threats: (1) email account compromise (EAC) attacks; (2) advanced attacks; and (3) spam attacks. In some embodiments, the threat detection platform employs a single ML model able to classify these different types of security threats. In other embodiments, the threat detection platform employs multiple ML models, each of which is able to classify a different type of security threat.

[00126] Figure 8A includes a high-level illustration of the detection architecture of a threat detection platform in accordance with some embodiments. Initially, the threat detection platform will determine that an event has occurred or is presently occurring. One example of an event is the receipt of an incoming email. As discussed above, the threat detection platform may be programmatically integrated with an email service employed by an enterprise so that all external emails (e.g., those received from an external email address and/or those transmitted to an external email address) and/or all internal emails (e.g., those sent from one employee to another employee) are routed through the threat detection platform for examination.

[00127] Then, the threat detection platform will perform an entity resolution procedure in order to identify the entities involved in the event. Generally, the entity resolution procedure is a multi-step process.

[00128] First, the threat detection platform will acquire information regarding the event. For example, if the event is the receipt of an incoming email, the threat detection platform may examine the incoming email to identify the origin, sender identity, sender email address, recipient identity, recipient email address, subject, header(s), body content, etc. Moreover, the threat detection platform may be able to determine whether the incoming email includes any links, attachments, etc. Figure 9 depicts an example of an incoming email that may be examined by the threat detection platform.

[00129] Second, the threat detection platform will resolve the entities involved in the event by examining the acquired information. Figure 10A depicts how information gleaned from the incoming email can be used to establish different entities (also referred to as "features" or "attributes" of the incoming email). Some information may correspond directly to an entity. Here, for example, the identity of the sender (or purported sender) may be established based on the origin or sender name. Other information may correspond indirectly to an entity. Here, for example, the identity of the sender (or purported sender) may be established by applying a natural language processing (NLP) algorithm and/or computer vision (CV) algorithm to the subject, body content, etc. Accordingly, entities may be established based on the incoming email, information derived from the incoming email, and/or metadata accompanying the incoming email. Figure 10B depicts an example of how the threat detection platform can execute the entity resolution procedure to establish the identity of the sender of the incoming email. Here, the threat detection platform has identified the sender identity based on (1) the sender name ("Bob Roberts") as derived from the incoming email and (2) the subject as processed by an NLP algorithm.

[00130] In some embodiments, the threat detection platform will augment the acquired information with human-curated content. For example, feature(s) of an entity may be extracted from human-curated datasets of well-known brands, domains, etc. These human-curated datasets may be used to augment information gleaned from the enterprise's own datasets. Additionally or alternatively, humans may be responsible for labeling entities in some situations. For example, a human may be responsible for labeling landing pages and/or Uniform Resource Locators (URLs) of links found in incoming emails. Human involvement may be useful when quality control is a priority, when comprehensive labeling of evaluation metrics is desired, etc. For example, a human may actively select which data/entities should be used for training the ML model(s) used by the threat detection platform.

[00131] The threat detection platform can then establish, compile, and/or calculate corpus statistics for the entities determined to be involved in the event. These corpus statistics can be stored/visualized in terms of an entity risk graph. As shown in Figure 11, the entity risk graph can contain historical combinations of these entities and risk scores associated with those historical combinations. Thus, the entity risk graph represents one way to visualize the types of corpus statistics that have been established, compiled, and/or calculated by the threat detection platform. Each node in the entity risk graph corresponds to a real-world entity, IP address, browser, etc. Accordingly, the entity risk graph may include a risk score for a domain detected in the incoming email, a risk score for an IP address detected in metadata accompanying the incoming email, a risk score for the sender ("Employee A") communicating with the recipient ("Employee B"), etc. Each connection between a pair of nodes, meanwhile, is representative of risk as determined on past events involving those nodes. Figure 12 depicts an example of an entity risk graph.

[00132] Figure 8B includes a more detailed example of a process by which a threat detection platform can process data related to past emails (here, acquired from Microsoft Office® 365), extract primary attributes from the past emails,

generate corpus statistics based on the primary attributes, derive secondary attributes based on the primary attributes and the corpus statistics, train ML model(s) with the primary attributes and/or the secondary attributes, and then employ the ML model(s) to score incoming emails based on the risk posed to an enterprise.

[00133] Figure 13 provides an example matrix of the stages that may be performed by a threat detection platform as it processes data, extracts features, determines whether an event is representative of an attack, etc. During a first stage, the threat detection platform can download various data related to the communication activity of an enterprise. For example, the threat detection platform may establish, via an API, a connection with a storage medium that includes data related to past communication activities involving employees of the enterprise. The storage medium could be, for example, an email server that includes past emails sent/received by employees of the enterprise. Accordingly, the threat detection platform can download various data into a local processing environment, such as raw emails, raw attachments, raw directory list(s) (e.g., the enterprise's Microsoft Outlook® directory), raw mail filters, raw risk events, etc.

[00134] During a second stage, the threat detection platform can extract text, metadata, and/or signals (collectively referred to as "extracted items") from the downloaded data. For example, the threat detection platform may identify an attachment signal in an email using learned model parameter(s) for text extraction. The term "extracted signal," as used herein, refers to any information, raw or derived, used by the algorithm(s) employed by the threat detection platform as input. Examples of extracted signals include, but are not limited to, structured data such as IP addresses, third-party data or datasets, API-based integration information with any third-party tool, or other enterprise data or datasets. The extracted items can be persisted in a columnar format in which each column is independently updated. As shown in Figure 14A, each column may be associated with one of three different criteria: (1) extractors (e.g., auth extraction); (2) model application (e.g., extracting spam-text model prediction);

and (3) rule (e.g., extract specific phrase defined via a rules interface). Figures 14B-C, meanwhile, depicts examples of data structures that may be populated by the threat detection platform using the extracted items.

[00135] During a third stage, the threat detection platform can identify the entities involved in a communication activity. For example, if the communication activity is the receipt of an incoming email, the threat detection platform may identify the sender identity, sender email address, or topic based on the text, metadata, and/or signals extracted during the second stage. As discussed above, humans may be responsible for assisting in entity resolution in some instances. Thus, the third stage may be partially performed by human(s) and partially performed by the threat detection platform, or may be entirely performed by the threat detection platform.

[00136] During a fourth stage, the threat detection platform can produce summaries for the entities (also referred to as "attributes" of the email) identified in the third stage based on the past communication activities involving those entities. That is, the threat detection platform may produce corpus statistics that are representative of risk scores associated with historical combinations of the entities identified in the third stage. These corpus statistics can be stored/visualized in terms of an entity risk graph, as shown in Figure 12. Additionally or alternatively, these corpus statistics can be stored in one or more databases. Figure 14D depicts an example of a database that includes all corpus statistics, and Figure 14E depicts an example of a database that includes the corpus statistics related to the sender.

[00137] During a fifth stage, the threat detection platform can produce a score indicative of risk to the enterprise. Scores may be produced on a per-communication, per-attack-type, or per-entity basis. Accordingly, the threat detection platform may score each incoming email directed to an employee of the enterprise to determine which incoming emails, if any, should be prevented from reaching the employee's inbox. Generally, incoming emails are scored based on

compromise score, and compromise is scored based on the number/type of malicious emails that are received. For example, the threat detection platform may include a threat detection engine and an account compromise engine that separately consider incoming emails. The output produced by each engine (e.g., in terms of score, suspiciousness, etc.) may be used by the other engine for better detection. For example, if an account is determined to be in the suspicious range by the account compromise engine, all emails originating from that account may be monitored more sensitively by the threat detection engine. This protects against cases where an unauthorized entity (also referred as an "attacker") takes over an account and then uses that account to mount phishing attacks. Scoring of communication activities is further discussed below with respect to Figure 21.

[00138] In some embodiments, the threat detection platform also "hydrates" the entities identified in the third stage. The term "hydrate," as used herein, refers to the action of appending additional signals to a communication such as an email. These additional signals can be defined in three locations: (1) code-defined extractor (e.g., secondary attributes); (2) model application (e.g., URL extraction model, ransom model, employee impersonation model); and (3) rule (e.g., specific whitelists or blacklists). As shown in Figure 14F, an email can be hydrated using a directed acyclic graph (DAG) of databases, rules, and/or models to produce a final set of signals to be used in detection. Figure 14G illustrates an example of a hydrated email (i.e., an email with primary, secondary, and/or scored attributes).

[00139] During a sixth stage, the threat detection platform can compare each score with a threshold to determine how the email should be classified. For example, the threat detection platform may determine whether to classify the email as borderline, suspicious, or bad. The thresholds used to determine how each email should be classified may be controlled using live data and/or replay from logs to determine the acceptable number of flagged messages. In some embodiments, the thresholds are updated, continually or periodically, to maintain a target flag rate. For example, the threat detection platform may alter the

threshold so that a predetermined percentage of all incoming emails (e.g., 0.1%, 0.5%, or 1.0%) are flagged as borderline, suspicious, or bad. The threshold for a given model may be calibrated based on an internal target for the number of false positives and/or false negatives generated by the given model. Generally, increasing a threshold will result in the model having fewer false positives at the expense of more false negatives, while lowering a threshold will result in the model having fewer false negatives at the expense of more false positives. Figure 14H illustrates how each rule/model employed by the threat detection platform may return a score that can be modulated by a threshold. These rules/models may correspond to a subset of the entities extracted in the second stage.

Threat Intelligence

[00140] Customers may desire to receive threat intelligence about attacks that the threat detection platform has discovered. Because the threat detection platform can monitor incoming emails in real time, unique threat intelligence can be produced that allows abnormal communication activities to be detected more quickly than would be possible with conventional email filtering services.

[00141] The threat detection platform can be designed to serve as a centralized system that captures indicators of compromise (IOC) gleaned from a variety of sources, including internal sources (e.g., enterprise databases) and external sources. Examples of IOCs include IP addresses, email addresses, URLs, domains, email attachments, cryptocurrency (e.g., Bitcoin) addresses, etc. A database of IOCs can be used for a few different purposes. While the most paramount purpose is the detection of incoming emails representative of security threats, the database could be provided to enterprises for ingestion into other security products, such as firewalls, security orchestration, automation and response (SOAR) tools, etc. For example, an enterprise may find it useful to provide IOCs deemed to be malicious to a management tool, such as a gateway, to help protect employees from future threats, poor choices, etc. As another

example, an enterprise may surface employee accounts associated with IOCs for further examination (e.g., to determine whether those employee accounts have been compromised). Additionally or alternatively, the threat management platform may be programmed to infer a threat condition of each IOC. For instance, the threat management platform might classify each IOC as being representative of a phishing, malware, or compromise operation.

[00142] Many enterprises may find it sufficient to examine malicious email campaigns and comprised employee accounts surfaced by the threat detection platform. However, some enterprises have begun monitoring IOCs in order to address security threats in real time. For instance, an enterprise may monitor IOCs gleaned from incoming emails by the threat detection platform to identify appropriate responses and/or proactive measures to prevent these IOCs from entering its environment again in the future.

[00143] At a high level, the threat detection platform can be designed to perform various tasks including:

- Ingestion of threat intelligence from different types of sources such as:
  - Inferred IOCs based on statistics of previously-seen attacks (e.g., the number of good or bad emails sent from the same source IP address);
  - IOCs based on detected attacks (e.g., compromised domains and phishing links); and
  - Internal security analysts installed by enterprises; and
- Exportation of threat intelligence (e.g., as a database for inline use in examining incoming emails, or as a feed to be ingested by other security threat products);

[00144] Embodiments of the threat detection platform may also be designed to permit enabling/disabling of IOCs on a per-enterprise basis. For example, an enterprise may upload a list of IOCs to the threat detection platform that should be used specifically when examining their emails. Moreover, the threat detection

platform may annotate IOCs with a probability so that those IOCs which are probably malicious can be supported. Thus, the threat detection platform could be designed to flag those emails determined to be malicious, as well as those emails that may be malicious. In some embodiments, the threat detection platform is able to place a time limit on each IOC to prevent permanent blacklisting. For example, if a given website is found to be hosting a phishing website, the threat detection platform may capture the given website as an IOC for a specified time period after which it checks whether the given website is still hosting the phishing website.

[00145] Figure 15A includes a high-level system diagram of a threat intelligence system of which the threat detection platform is a part. As shown in Figure 15, IOCs can be produced/identifier by several different sources. These sources include incoming email, URLs, domains, external feeds (e.g., from another security threat product), internal security analysts, etc.

[00146] The threat detection platform may overlay the IOCs with attacks discovered, for example, by examining incoming email. That is, the threat detection platform may attempt to match the IOCs with corresponding attacks so that the score calculated for each attack can be attributed to the appropriate IOC. Thereafter, the threat detection platform may filter the IOCs (e.g., based on the scores that have been attributed to them) and then use the filtered IOCs (and corresponding scores) to further bolster the ability to detect security threats.

[00147] In some embodiments, the threat detection platform may utilize its ecosystem of multiple enterprises to offer federated capabilities. For example, the threat detection platform could build a central vendor database across its entire environment to establish a list of vendors and learn what constitutes normal behavior for each vendor. For example, the central vendor database may specify the email endpoints used by each vendor, the accountant(s) responsible for sending invoices for each vendor, the invoicing software used by each vendor, the routing/bank account numbers for each vendor, the location from

which the invoices of each vendor originate, etc. As another example, the threat detection platform could build a central threat database across its entire environment to establish a list of entities (e.g., IP addresses, URLs, domains, email addresses) that are most notable in sending attacks. The central threat database may be helpful as it permits the threat detection platform to apply knowledge gained from one enterprise across the entire ecosystem. As another example, the threat detection platform may automatically monitor inboxes to which employees have been instructed to forward suspicious email. When the threat detection platform discovers a malicious email that was missed by its ML models, the threat detection platform may automatically pull the malicious email from all other inboxes in the enterprise where it was found. Moreover, the threat detection platform may use its federated ecosystem to pull the malicious email from inboxes of other enterprises.

[00148] Generally, the threat detection platform is designed so that datasets can be computed, tracked, and added to the modeling pipeline in which ML models are developed, trained, etc. Each dataset may be readily reproducible, updatable, and searchable/viewable. As noted above, the datasets may be editable through interfaces generated by the threat detection platform. For example, a human may label different elements included in a dataset for the purpose of training a ML model. Examples of databases that may be accessible to the threat detection platform include:

- A vendor database that includes a set of common vendors from which enterprises receive emails. Examples of vendors include American Express®, Chase®, Lloyd's Bank®, Microsoft®, etc. In the vendor database, each vendor may be associated with a canonical name, a list of safe domains (e.g., domains that an email links to, domains the emails are received from, domains with which the vendor works), a list of alias names, a list of regular expressions (e.g., "Employee via Third-Party Service"), or another appropriate signifier, etc. The threat detection platform may use the vendor database to whitelist domains

known to be good/safe that vendors send emails from or to perform other types of email scoring or analysis.

- A domain database that includes a set of top-level domains. For each domain, the threat detection platform can track some additional data. For example, the threat detection platform may establish whether the domain has been whitelisted as a safe domain, whether the domain corresponds to a hosting service, and whether the domain is a redirector. Moreover, the threat detection platform may establish what, if anything, Google's SafeBrowsing API says about the domain, how often the domain is included in emails received by the enterprise, how much labeled data can be seen, what cached Whois data is available for the domain, etc.
- A Whois registrant database that includes information about each registrant that is derived from Whois data stored in the domain database.
- A URL database that includes URL-level data derived from links included in emails received an enterprise. For each URL, the threat detection platform may populate an entry with a model of URL suspiciousness, external data regarding URL quality (e.g., data from phishtank.com), data acquired via Google's SafeBrowsing API, or statistics regarding how often the URL is seen by the enterprise.
- An employee database that include information on employees of an enterprise. Generally, the threat detection platform maintains a separate employee database for each enterprise whose email is being monitored. For each employee, the threat detection platform may populate an entry with a company identifier, name, employee identifier, alias name(s), common email address(es) (e.g., enterprise email addresses and personal email addresses that have been verified), Lightweight Directory Access Protocol (LDAP) role, and the number of attacked observed against the employee.

- A label database (also referred to as a "feedback database") that includes labelled data to be used to build aggregated feedback for each enterprise, employee, etc. An entry could include aggregated feedback for an email address, domain, link, normalized/hashed body, etc. For example, an entry in the label database may specify that 15 of 30 labels for emails from "A@exploit.com" have been labeled as positive for attacks, or that 10 out of 11 labels for emails containing a link to http://xyz.com have been labeled as positive for attacks.

[00149] As discussed above, an enterprise may monitor IOCs gleaned from incoming emails by the threat detection platform to identify appropriate responses and/or proactive measures to prevent these IOCs from entering its environment again in the future. By exposing IOCs in a rapid manner, the threat detection platform can alert enterprises so that security postures can be improved to counteract security threats. Figure 15B depicts an example of an interface through which an enterprise can examine IOCs discovered by the threat detection platform.

[00150] In some embodiments, the threat detection platform provides the ability to extract and/or export IOCs. For example, through the interface shown in Figure 15B, an enterprise may export information (also referred to as "threat intelligence") related to these IOCs into a management tool to improve its ability to detect/address these security threats in the future. The threat detection platform may format the information (e.g., into a machine-readable form) so that it is readable shareable. For example, the information may be formatted in accordance with the Structured Threat Information Expression (STIX) and Trusted Automated Exchange of Indicator Information (TAXII) specifications. Generally, STIX will indicate what type of threat intelligence is formatted, while TAXII will define how the underlying information is relayed.

[00151] A schema may be employed to ensure that threat intelligence is accounted for in a consistent manner. For a given IOC, the scheme may indicate:

- An observable output (e.g., the actual URL, IP address, domain, or account);
- A classification (e.g., whether the IOC is private or public);
- A type (e.g., whether the IOC is a URL, IP address, domain, or account);
- A severity (e.g., whether the IOC poses a low, medium, high, or very high threat);
- A confidence metric (e.g., a score on a 0-100 scale indicating confidence that the IOC represents a security threat);
- An observed time; and/or
- A Traffic Light Protocol (TLP) metric indicating how widely the underlying information should be shared.

As shown in Figure 15B, some of this data may be presented on the interface for review by the enterprise. For example, the interface may allow the enterprise to readily sort IOCs by severity level so that those IOCs representing the largest threat can be dealt with.

[00152] Figure 16 illustrates how a threat detection platform may derive/infer attributes from data acquired from various sources, profile those attributes to ML models as input, and then examine the output produced by those ML models to determine whether a security threat exists. As shown in Figure 16, the attributes could be provided as input to various ML models associated with different types of attacks. Here, for example, features related to the body style (e.g., HTML, signature, phone number, etc.) of an incoming email could be fed to ML models designed to detect internal employee EAC attacks, system EAC attacks, external EAC attacks, employee impersonation attacks, vendor impersonation attacks, and partner impersonation attacks.

[00153] Figure 17 includes a high-level architectural depiction of a threat detection platform able to perform generate/update the data used for real-time processing of incoming emails via batch processing. Batch processing may be particularly helpful in facilitating real-time processing to further enhance the threat detection capabilities of the threat detection platform. This concept may be referred to as near real-time scoring, and it could be used for compute-intensive detection tasks such as processing the attachments that have been attached to incoming emails.

[00154] Threat intelligence may represent a core pillar of a long-term strategy for addressing email-based security threats. For example, a threat detection platform may be employed by an enterprise to better understand threats to its security in several ways. First, the threat detection platform could examine corpus statistics to detect instances of employee account compromise (EAC). For instance, given a series of sign-in activities and email activities, how often are good events and/or bad events detected for a particular attribute such as IP address, sender email address, sender location, etc. Second, the threat detection platform could examine corpus statistics to determine what constitutes normal/abnormal communication activity based on the attributes of emails associated with an enterprise. Third, the threat detection platform could produce a set of "bad entities" or "malicious entities" that enterprises can programmatically access to trigger actions in their respective environments. For instance, an enterprise may configure its firewall based on the set of bad entities. Examples of entities include employees, brands, vendors, domains, locations, etc. Fourth, the threat detection platform could generate and/or react to signatures considered malicious in near real time (e.g., within minutes) of obtaining the necessary data. Fifth, given an attribute of a risk event, the threat detection platform could identify past risk events that contained this attribute. By analyzing these past risk events, the threat detection platform can better understand whether the attribute is associated with risk events that were ultimately determined to be safe or malicious. A specialized module (also

referred to as a "graph explorer") may be responsible for visually displaying how these past risk events affect the determination of riskiness.

[00155] At a high level, the threat detection platform can be described as analyzing risk events (or simply "events") to discover threats to enterprises. An example of a risk event is the receipt or transmission of an email. Another example of a risk event is a sign-in activity or some other communication with a cloud-based mail provider. Another example of a risk event is the creation of a mail filter. The maliciousness of a given risk event may be correlated with the maliciousness of the entities associated with the given risk event. For example, a mail filter will correspond to an employee of an enterprise, an email with an invoice will be received from a vendor, etc. All of these entities are connected to each other by arbitrary connections (e.g., a sender of the email will work for the vendor, the employee will email other employees of the enterprise). The term "signature," as used herein, refers to the combination of one or more attributes that categorizes a risk event. Signatures may be key to counting the risk events that have a particular combination of attributes.

[00156] Figure 18A includes a high-level illustration of a process by which a threat detection platform can perform threat intelligence. As shown in Figure 18A, data can be obtained from several different inputs (also referred to as "sources"). Here, configuration data includes definitions of risk event attributes that will be tracked by the threat detection platform. For example, the configuration data may include an instruction/algorithm that prompts the threat detection platform to "listen" for risk events that are associated with a given display name and a given sender email address. Domain-specific raw data (e.g., incoming emails with attributes) can also be obtained by the threat detection platform. In some embodiments, a user is permitted to provide functions that extract/map risk events to their attributes.

[00157] The event ingester module (or simply "event ingester") may be responsible for converting the raw data into an internal schema for risk events.

The schema may be designed to hold various risk events regardless of type (e.g., email, sign-in activity, mail filter). The stats builder module (or simply "stats builder") may be responsible for mapping signatures of attribute dimensions for a date range to counts of risk events.

[00158] Figure 18B includes a high-level illustration of a process by which a threat detection platform can "productionalize" a signature to be used to determine the threat posed by incoming emails. Initially, the real-time scoring module (also referred to as the "RT scorer") can process raw data related to incoming emails. The processed data associated with each incoming email can be passed to a counting service (also referred to as a "counting system") that converts the processed data into a processed risk event.

[00159] Moreover, each incoming email labeled through the frontend (e.g., via an interface generated by the threat detection platform) can be passed to the counting service that converts the labeled email into a processed risk event. The labels may indicate whether the incoming emails represent a security threat. Accordingly, the processed risk events derived from the labeled emails may be associated with a security risk metric.

[00160] The processed risk events created by the counting service can be stored in a database (e.g., a Redis distributed database). This database can be queried for signatures. For example, a query may be submitted for a whitelist of signatures determined not to represent a security threat. As another example, a query may be submitted for a count of signatures having a given attribute or combination of attributes.

Instant Signatures and Corpus Statistics

[00161] As discussed above, embodiments of the threat detection platform can be designed to detect security threats by examining behavior, identity, and content rather than metadata, links, domains, signatures, etc. However, it may be advantageous to consider this information in order to detect security threats in

a more accurate, consistent, and efficient (e.g., in terms of time and resources) manner.

[00162] Several different components of the threat detection platform could extract value from this information. Examples of such components include:

- A database that ingests signatures on a periodic basis and then uses these signatures to detect attacks;
- A ML model designed to ingest signatures on a periodic basis and then employ those signatures in a probabilistic fashion to detect attacks;
- An algorithm able to aggregate activities deemed to be safe or normal over history to be provided as inputs to ML models;
- A trawling module (also referred to as a "trawler") able to create new signatures by examining raw data from the past;
- A ML model designed to infer a general rule for detecting URL-based attacks by examining past emails with unsafe URLs; and
- A ML model designed to periodically examine signatures (or the raw data from which those signatures are derived) to detect changes in communication patterns (e.g., as determined based on the subject line, sender address, etc.).

[00163] For example, assume that an email is received from a previously unseen address (attacker@badsite1.com), and the email includes an attachment with a link to an unknown website (badsite2.net). The threat detection platform may separately and immediately identify all emails received from this address as potentially representing a security threat and all emails including a link to this website as potentially representing a security threat. This could be done without requiring that a human review the unknown website.

[00164] The keys to accomplishing this include (1) updating corpus statistics in a timely (i.e., non-batch) manner and (2) having date-indexed corpus statistics. Figure 19A includes a high-level illustration of a process by which a threat

detection platform can index corpus statistics to create a date-partitioned database of signatures and corpus statistics that can be used to identify unsafe entities more effectively. Such a process allows the unsafe entities to be identified via an exact match with signatures residing in the database and a probabilistic match with the signatures as determined by the ML models. Moreover, this bi-furcated approach to identifying unsafe entities allows the threat detection platform to react more quickly to attacks that involve unsafe domains and possibility unsafe domains (also referred to as "compromised domains"), as well as attacks that leverage safe domains such as dropbox.com.

[00165] Figure 19B depicts an example of a database that includes signatures and corpus statistics. The date partition may be needed so that the database can be used against messages without future knowledge in training. The database may be updated in near real time based on, for example, outputs produced by a real-time scoring module (e.g., RT scorer of Figure 18B) and/or labels input via an interface (e.g., as received by frontend of Figure 18B). As discussed above, the database may be populated/backfilled based on past emails associated with a given interval of time (e.g., 3, 6, 9, or 12 months).

[00166] Conceptually, the threat detection platform can organize data into one or more data structures. In the case of corpus statistics, for example, each enterprise may be assigned a single table. These tables may have "N" number of rows, where N is a relatively fixed integer. For example, a table for corpus statistics may include 270 rows if the threat detection platform is interested in tracking 270 days' worth of data, 365 rows if the threat detection platform is interested in tracking 365 days' worth of data, etc. Similarly, the threat detection platform could assign a single table for signatures to each enterprise. However, the number of rows in these tables will normally vary as new signatures are discovered in incoming emails.

Employee Account Compromise

[00167] Figure 20 illustrates how the threat detection platform can detect instances of employee account compromise (EAC). At a high level, the threat detection platform can learn about an enterprise by identifying their launch points (e.g., virtual private networks (VPNs) and IP addresses), establishing which launch points are considered normal, and then employing personalized, enterprise-based learning to detect security threats. Here, for example, the threat detection platform examines raw data (e.g., in the form of mail filters, sign ins, risk events, and phishing messages) and aggregated data (e.g., in the form of corpus statistics, sign-in corpus statistics, and auxiliary databases) to discover one or more user compromise signals.

[00168] Then, the threat detection platform employs multiple detectors to score the user compromise signals. Each score may be representative of how highly the user compromise signal corresponds to the likelihood that an employee's account has been compromised. Accordingly, the user compromise signals may be discovered on a per-user basis (e.g., for each employee of an enterprise).

[00169] The threat detection platform can detect instances of EAC by comparing user activity to the scored user compromise signals. For example, the threat detection platform may discover, based on the location and/or the frequency of sign ins, that a given user's account may have become compromised. However, the threat detection platform need not necessarily take action immediately. For instance, the threat detection platform may determine what actions, if any, to take based on which user compromise signal(s) indicate abnormal behavior, the score(s) of those user compromise signal(s), etc. As an example, the threat detection platform may take immediate action to prevent further accesses of the account if the relevant user compromise signal(s) have high scores, but the threat detection platform may continue to monitor the account if the relevant user compromise signal(s) have low scores.

Methodologies for Accurate Scoring

[00170] The term "accurate scoring" covers a combination of several concepts further discussed above. Figure 21 depicts a high-level flow diagram of a process 2100 for scoring the threat posed by an incoming email.

[00171] First, a threat detection platform may employ one or more ML models, such as deep learning models, to consume the universe of features that have been extracted for an incoming email from primary and secondary attributes to identify potential security threats (step 2101). Collectively, these ML model(s) may be referred to as the "ML detector." In some embodiments, a real-time proportional-integral-derivative (PID) controller is used to tune the threshold for each entity whose emails are being monitored to take into consideration the changing landscape of attack type, email content, etc. The thresholds ensure that the ML model(s) have high precision and continue to be highly precise over time. To cover the general attack landscape, the threat management platform may employ a combination of federated ML models and enterprise-specific ML models able to capture the per-enterprise nuances of sophisticated attacks (e.g., spear phishing attacks).

[00172] Second, the threat detection platform can glean the signatures of IOCs in real time to determine the nature of any security threats identified by the ML detector (step 2102). Examples of IOCs include IP addresses, email addresses, URLs, domains, etc. For zero-hour attacks, as email-based attacks are identified by the ML detector, the IOCs can be extracted from the emails. These IOCs can be automatically ingested into database as "signatures" in real time. Thereafter, the signatures can be used in conjunction with the ML detector to discover future attacks with the same features.

[00173] Third, the threat detection platform can perform deep feature extraction to identify zero-hour attacks (step 2103). Identifying zero-hour attacks requires deeper content analysis to understand the nuances of a possible attack. For example, deep learning sub-model(s) may be applied to understand the text,

content, sentiment, and/or tone of an email. As another example, to find phishing pages, computer vision may be used to compare landing page of a link embedded in an email to a set of known sign-on pages. As another example, webpage crawling may be performed to extract information regarding a deep link (e.g., a link embedded in an attachment or a link accessible on a linked website) to discover instances of deep phishing.

Threat Detection, Assessment, and Remediation

[00174] Figure 22 depicts a flow diagram of a process 2200 for applying a personalized machine learning (ML) model to emails received by an employee of an enterprise to detect security threats. Initially, a threat detection platform can establish a connection with a storage medium that includes first data related to past emails received by the employee of the enterprise (step 2201). The first data could include the past emails themselves, or information related to the past emails such as primary attributes or secondary attributes. In some embodiments, the threat detection platform establishes the connection with the storage medium via an application programming interface (API). In such embodiments, the threat detection platform may not establish the connection with the storage medium until it receives input indicative of an approval from an administrator associated with the enterprise to access the first data.

[00175] The threat detection platform can download a first portion of the first data into a local processing environment (step 2202). For example, the threat detection platform may download all emails received by the employee over the last 3 months, 6 months, 12 months, etc. Then, the threat detection platform can build a personalized ML model for the employee based on the first portion of the first data (step 2203). For example, the threat detection platform may parse each email included in the first data to discover one or more attributes, and then the threat detection platform can provide these attributes to the ML model as input for training. Examples of attributes include the sender name, sender email address, subject, etc. Because the personalized ML model is trained using past

emails received by the employee, normal communication habits can be established immediately upon deployment.

[00176] Thereafter, the threat detection platform may receive second data related to an email received by the employee (step 2204). The threat detection platform can establish whether the email is representative of a security risk by applying the personalized ML model to the second data (step 2205). Such action will cause the personalized ML model to produce an output that indicates whether the email is indicative of an attack. For example, the output may specify whether the email is suspicious, or the output may specify that the email does not conform with the employee's past communication activities.

[00177] In some instances, the threat detection platform will determine, based on the output, that the email is indicative of an attack (step 2206). In such instances, the threat detection platform can characterize the attack along multiple dimensions (also referred to as "facets") (step 2207). For example, the threat detection platform may establish the identity of the attacked party, the attack vector, the identity of the impersonated party, the impersonation strategy, and/or the attack goal.

[00178] Other steps may also be included in some embodiments. For example, the threat detection platform may download a second portion of the first data corresponding to the local processing environment. The second portion of the first data may correspond to a different interval of time than the first portion of the first data. For example, the first portion of the first data may include all emails received by the employee over the last 6 months, and the second portion of the first data may include all emails received by the employee from 6-12 months ago. Then, the threat detection platform can establish whether any emails included in the second portion of the first data are representative of security risks by applying the personalized ML model to the second portion of the first data.

[00179] Figure 23 depicts a flow diagram of a process 2300 for detecting and characterizing email-based security threats in real time. Initially, a threat detection platform can receive an email addressed to an employee of an enterprise (step 2301). Then, the threat detection platform can apply a first model to the email to produce a first output indicative of whether the email is representative of a malicious email (step 2302). The first model may be trained using past emails addressed to the employee (and possible other employees) that have been verified as non-malicious emails. Accordingly, the first model may be referred to as the "surely-safe model." The first model serves as the first level of threat detection, and therefore may be tuned/designed to permit most email (e.g., upwards of 90, 95, or 99 percent of all incoming email) to reach the intended destination.

[00180] In some instances, the threat detection platform will determine, based on the first output, that the email is representative of a malicious email (step 2303). In such instances, the threat detection platform can apply a second model to the email to produce a second output indicative of whether the email is representative of a given type of malicious email (step 2304). As discussed above, the second model may be one or multiple models applied to the email responsive to determining that the email is representative of a malicious email. Thus, the threat detection platform may apply multiple models to the email to produce multiple outputs, and each model of the multiple models may correspond to a different type of malicious email.

[00181] The threat detection platform can then determine whether to remediate the email based on the second output (step 2305). That is, the threat detection platform can determine what actions, if any, should be performed based on the second output. For example, if the second output indicates that the email includes a link to an HTML resource, the threat detection platform may follow the link so that the HTML resource is accessed using a virtual web browser, extract a Document Object Model (DOM) for the HTML resource through the virtual web browser, and analyze the DOM to determine whether the link represents a

security risk. As another example, if the second output indicates that the email includes a primary link to a resource hosted by a network-accessible hosting service (e.g., Google Drive®, Box®, Dropbox®, or Microsoft OneDrive®), the threat detection platform may follow the primary link so that the resource is accessed using a virtual web browser, discover whether any secondary links to secondary resources are present by examining content of the resource through the virtual web browser, follow each secondary link to analyze content of the corresponding secondary resource through the virtual web browser, and determine whether the primary link represents a security threat based on whether any secondary links were determined to represent security threats. As another example, if the second output indicates that the email includes a link to an HTML resource, the threat detection platform may follow the link so that the HTML resource is accessed using a virtual web browser, capture a screenshot of the HTML resource through the virtual web browser, apply a computer vision (CV) algorithm designed to identity similarities between the screenshot and a library of verified sign-in websites, and determine whether the link represents a security threat based on an output produced by the CV algorithm. As another example, if the second output indicates that the email includes an attachment, the threat detection platform may open the attachment within a secure processing environment and then determine whether the attachment represents a security threat based on an analysis of content of the attachment. For instance, the threat detection platform may use a headless browser instance running on a standalone computer server (also referred to as a “sandbox computer server”) to inspect the attachment (e.g., by generating a screenshot of its contents), rather than open the attachment directly on a computing device associated with the recipient of the email. Moreover, the threat detection platform may examine any links included in the attachment as discussed above.

[00182] In some embodiments, the threat detection platform is further configured to apply a third model designed to produce a comprehensible visualization component based on the second output (step 2306). In embodiments where the second model is part of an ensemble of models applied

by the threat detection platform, the third model can aggregate the outputs produced by the models in the ensemble, characterize the attack based on the aggregated outputs, and then convert the aggregated outputs into a comprehensible visualization component. For example, the third model may generate a notification that identifies the type of security threat posed by the email, whether remediation actions are necessary, etc. The comprehensible visualization component may be created so that security professionals responsible for addressing/mitigating security threats can more easily understand why an incoming email was flagged as being representative of an attack.

Processing System

[00183] Figure 24 is a block diagram illustrating an example of a processing system 2400 in which at least some operations described herein can be implemented. For example, some components of the processing system 2400 may be hosted on a computing device that includes a threat detection platform (e.g., threat detection platform 214 of Figure 2). As another example, some components of the processing system 2400 may be hosted on a computing device that is queried by a threat detection platform to acquire emails, data, etc.

[00184] The processing system 2400 may include one or more central processing units ("processors") 2402, main memory 2406, non-volatile memory 2410, network adapter 2412 (e.g., network interface), video display 2418, input/output devices 2420, control device 2422 (e.g., keyboard and pointing devices), drive unit 2424 including a storage medium 2426, and signal generation device 2430 that are communicatively connected to a bus 2416. The bus 2416 is illustrated as an abstraction that represents one or more physical buses and/or point-to-point connections that are connected by appropriate bridges, adapters, or controllers. The bus 2416, therefore, can include a system bus, a Peripheral Component Interconnect (PCI) bus or PCI-Express bus, a HyperTransport or industry standard architecture (ISA) bus, a small computer system interface (SCSI) bus, a universal serial bus (USB), IIC (I2C) bus, or an Institute of

Electrical and Electronics Engineers (IEEE) standard 1394 bus (also referred to as "Firewire").

[00185] The processing system 2400 may share a similar computer processor architecture as that of a desktop computer, tablet computer, personal digital assistant (PDA), mobile phone, game console, music player, wearable electronic device (e.g., a watch or fitness tracker), network-connected ("smart") device (e.g., a television or home assistant device), virtual/augmented reality systems (e.g., a head-mounted display), or another electronic device capable of executing a set of instructions (sequential or otherwise) that specify action(s) to be taken by the processing system 2400.

[00186] While the main memory 2406, non-volatile memory 2410, and storage medium 2426 (also called a "machine-readable medium") are shown to be a single medium, the term "machine-readable medium" and "storage medium" should be taken to include a single medium or multiple media (e.g., a centralized/distributed database and/or associated caches and servers) that store one or more sets of instructions 2428. The term "machine-readable medium" and "storage medium" shall also be taken to include any medium that is capable of storing, encoding, or carrying a set of instructions for execution by the processing system 2400.

[00187] In general, the routines executed to implement the embodiments of the disclosure may be implemented as part of an operating system or a specific application, component, program, object, module, or sequence of instructions (collectively referred to as "computer programs"). The computer programs typically comprise one or more instructions (e.g., instructions 2404, 2408, 2428) set at various times in various memory and storage devices in a computing device. When read and executed by the one or more processors 2402, the instruction(s) cause the processing system 2400 to perform operations to execute elements involving the various aspects of the disclosure.

[00188] Moreover, while embodiments have been described in the context of fully functioning computing devices, those skilled in the art will appreciate that the various embodiments are capable of being distributed as a program product in a variety of forms. The disclosure applies regardless of the particular type of machine or computer-readable media used to actually effect the distribution.

[00189] Further examples of machine-readable storage media, machine-readable media, or computer-readable media include recordable-type media such as volatile and non-volatile memory devices 2410, floppy and other removable disks, hard disk drives, optical disks (e.g., Compact Disk Read-Only Memory (CD-ROMS), Digital Versatile Disks (DVDs)), and transmission-type media such as digital and analog communication links.

[00190] The network adapter 2412 enables the processing system 2400 to mediate data in a network 2414 with an entity that is external to the processing system 2400 through any communication protocol supported by the processing system 2400 and the external entity. The network adapter 2412 can include a network adaptor card, a wireless network interface card, a router, an access point, a wireless router, a switch, a multilayer switch, a protocol converter, a gateway, a bridge, bridge router, a hub, a digital media receiver, and/or a repeater.

[00191] The network adapter 2412 may include a firewall that governs and/or manages permission to access/proxy data in a computer network, and tracks varying levels of trust between different machines and/or applications. The firewall can be any number of modules having any combination of hardware and/or software components able to enforce a predetermined set of access rights between a particular set of machines and applications, machines and machines, and/or applications and applications (e.g., to regulate the flow of traffic and resource sharing between these entities). The firewall may additionally manage and/or have access to an access control list that details permissions including the

access and operation rights of an object by an individual, a machine, and/or an application, and the circumstances under which the permission rights stand.

[00192] The techniques introduced here can be implemented by programmable circuitry (e.g., one or more microprocessors), software and/or firmware, special-purpose hardwired (i.e., non-programmable) circuitry, or a combination of such forms. Special-purpose circuitry can be in the form of one or more application-specific integrated circuits (ASICs), programmable logic devices (PLDs), field-programmable gate arrays (FPGAs), etc.

Remarks

[00193] The foregoing description of various embodiments of the claimed subject matter has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the claimed subject matter to the precise forms disclosed. Many modifications and variations will be apparent to one skilled in the art. Embodiments were chosen and described in order to best describe the principles of the invention and its practical applications, thereby enabling those skilled in the relevant art to understand the claimed subject matter, the various embodiments, and the various modifications that are suited to the particular uses contemplated.

[00194] Although the Detailed Description describes certain embodiments and the best mode contemplated, the technology can be practiced in many ways no matter how detailed the Detailed Description appears. Embodiments may vary considerably in their implementation details, while still being encompassed by the specification. Particular terminology used when describing certain features or aspects of various embodiments should not be taken to imply that the terminology is being redefined herein to be restricted to any specific characteristics, features, or aspects of the technology with which that terminology is associated. In general, the terms used in the following claims should not be construed to limit the technology to the specific embodiments disclosed in the specification, unless those terms are explicitly defined herein. Accordingly, the

actual scope of the technology encompasses not only the disclosed embodiments, but also all equivalent ways of practicing or implementing the embodiments.

[00195] The language used in the specification has been principally selected for readability and instructional purposes. It may not have been selected to delineate or circumscribe the subject matter. It is therefore intended that the scope of the technology be limited not by this Detailed Description, but rather by any claims that issue on an application based hereon. Accordingly, the disclosure of various embodiments is intended to be illustrative, but not limiting, of the scope of the technology as set forth in the following claims.

[00196] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[00197] The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as an acknowledgment or admission or any form of suggestion that that prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

Examples of Embodiments

[00198] Examples are provided here for the purpose of illustration only. Those skilled in the art will recognize that, unless contrary to physical possibility, each example could be combined any other example.

1. A computer-implemented method comprising:

establishing, via an application programming interface, a connection with a storage medium that includes a series of past communications received by an employee of an enterprise;

downloading, via the application programming interface, a first portion of the series of past communications corresponding to a first interval of time into a local processing environment;

building a machine learning (ML) model for the employee by providing the first portion of the series of past communications to the ML model as training data;

receiving, via the application programming interface, a communication addressed to the employee; and

establishing whether the communication represents a security risk by applying the ML model to the communication.

2. The computer-implemented method of example 1, further comprising:

receiving input indicative of an approval from an administrator associated with the enterprise to access the series of past communications;

wherein said establishing is performed in response to receiving the input.

3. The computer-implemented method of example 1, wherein the series of past communications includes multiple emails that were delivered to the employee.

4. The computer-implemented method of example 1, further comprising:

examining each past communication in the first portion of the series of past communications to establish an attribute; and

providing the attributes derived from the first portion of the series of past communications to the ML model as training data.

5. The computer-implemented method of example 1, further comprising:

determining, based on an output produced by the ML model, that the communication represents a security risk; and

characterizing the security risk along multiple dimensions.

6. The computer-implemented method of example 5, wherein the multiple dimensions include:

an attacked party,

an attack vector,

an impersonated party,

an impersonation strategy, and

an attack goal.

7. The computer-implemented method of example 1, wherein the storage medium is a computer server managed by an entity other than the enterprise.

8. The computer-implemented method of example 1, wherein the first portion of the series of past communications includes all emails received by the employee during the first interval of time.

9. The computer-implemented method of example 1, further comprising:

downloading, via the application programming interface, a second portion of the series of past emails corresponding to a second interval of time that precedes the first interval of time into the local processing environment; and

establishing whether any emails received during the second interval of time represent security risks by applying the ML model to the second portion of the series of past emails.

10. The computer-implemented method of example 1, further comprising:

examining the communication to establish multiple attributes;

generating a statistical profile for the communication,

wherein the statistical profile includes a risk score for each pair of attributes included in the multiple attributes, each risk score being based on riskiness of historical communications involving the corresponding pair of attributes.

11. A non-transitory computer-readable medium with instructions stored thereon that, when executed by a processor, cause the processor to perform operations comprising:

receiving an email addressed to an employee of an enterprise;

applying a first model to the email to produce a first output indicative of whether the email is representative of a non-malicious email,

wherein the first model is trained using past emails addressed to the employee that have been verified as non-malicious emails;

determining, based on the first output, that the email may be a malicious email;

applying a second model to the email to produce a second output indicative of whether the email is representative of a given type of malicious email; and

performing an action with respect to the email based on the second output.

12. The non-transitory computer-readable medium of example 11,

wherein the second output indicates that the email is not a malicious email of the given type, and

wherein performing the action comprises:

forwarding the email to an inbox of the employee.

13. The non-transitory computer-readable medium of example 11, wherein the second model is one of multiple models applied to the email responsive to determining that the email may be a malicious email.

14. The non-transitory computer-readable medium of example 13, wherein each model of the multiple models is associated with a different type of malicious email.

15. The non-transitory computer-readable medium of example 14, wherein the multiple models produce multiple outputs when applied to the email, and wherein the operations further comprise:

applying a third model designed to aggregate the multiple outputs produced by the multiple models into a comprehensible visualization component.

16. The non-transitory computer-readable medium of example 11,

wherein the second output indicates that the email includes a link to a Hypertext Markup Language (HTML) resource, and

wherein performing the action comprises:

following the link so that the HTML resource is accessed using a virtual web browser,

extracting a Document Object Model (DOM) for the HTML resource through the virtual web browser, and

analyzing the DOM to determine whether the link represents a security threat.

17. The non-transitory computer-readable medium of example 11,

wherein the second output indicates that the email includes a primary link to a resource hosted by a network-accessible hosting service, and

wherein performing the action comprises:

following the primary link so that the resource is accessed using a virtual web browser,

discovering whether any secondary links to secondary resources are present by examining content of the resource through the virtual web browser,

for each secondary link,

following the secondary link so that the corresponding secondary resource is accessed using the virtual web browser, and

analyzing content of the corresponding secondary resource to determine whether the secondary link represents a security threat, and

determining whether the primary link represents a security threat based on whether any secondary links were determined to represent security threats.

18. The non-transitory computer-readable medium of example 11,

wherein the second output indicates that the email includes a link to a Hypertext Markup Language (HTML) resource, and

wherein performing the action comprises:

following the link so that the HTML resource is accessed using a virtual web browser,

capturing a screenshot of the HTML resource through the virtual web browser,

applying a computer vision algorithm designed to identify similarities between the screenshot and a library of verified sign-in websites, and

determining whether the link represents a security threat based on an output produced by the computer vision algorithm.

19. The non-transitory computer-readable medium of example 12,

wherein the second output indicates that the email includes an attachment, and

wherein performing the action comprises:

opening the attachment within a secure processing environment, and

determining whether the attachment represents a security threat based on an analysis of content of the attachment.

20. A computer-implemented method comprising:

receiving input indicative of an approval to access past emails that were delivered to an employee of an enterprise over a given interval of time;

establishing, via an application programming interface, a connection with a storage medium that includes the past emails;

downloading, via the application programming interface, the past emails into a local processing environment; and

building a machine learning (ML) model for identifying abnormal communication activity by providing the past emails to the ML model as training data.

21. The computer-implemented method of example 20, further comprising:

examining each past email downloaded into the local processing environment to identify a sender identity and a sender email address; and

populating entries in a database such that sender identities are associated with the corresponding sender email addresses as identified in the past emails.

22. The computer-implemented method of example 21, further comprising:

receiving an email addressed to the employee;

examining the email to establish a sender identity and a sender email address; and

determining whether the email represents a security threat based on whether the sender identity and the sender email address as identified in the email match an entry in the database.

23. The computer-implemented method of example 20, further comprising:
receiving an email addressed to the employee; and
establishing whether the email is indicative of an abnormal communication activity by applying the ML model to the email.

24. The computer-implemented method of example 23, wherein an output produced by the ML model upon being applied to the email indicates that the email message is indicative of an abnormal communication activity due to a presence of a previously-unknown sender identity, a previously-unknown sender email address, or a previously-unknown combination of sender identity and sender email address.

25. The computer-implemented method of example 23, further comprising:
responsive to a determination that the email is indicative of an abnormal communication activity,
uploading information related to the email to a federated database used to protect multiple enterprises against security threats.

26. A non-transitory computer-readable medium with instructions stored thereon that, when executed by a processor, cause the processor to perform operations comprising:
collecting data related to incoming emails and/or outgoing emails of a customer corresponding to a past interval of time;
generating a communication profile for the customer based on the data;
receiving an incoming email addressed to the customer;
deriving one or more attributes of the incoming email; and

determining whether the incoming email deviates from past email activity by comparing the one or more attributes to the communication profile.

27. The non-transitory computer-readable medium of example 26, wherein the customer is an enterprise for which the communication profile is generated.

28. The non-transitory computer-readable medium of example 26, wherein the customer is an employee of an enterprise for whom the communication profile is generated.

29. The non-transitory computer-readable medium of example 26, wherein said generating comprises:

deriving at least one attribute from each email corresponding to the past interval of time; and

building the communication profile based on the derived attributes.

30. The non-transitory computer-readable medium of example 26, the operations further comprising:

providing deviations in the incoming email to a machine learning (ML) model as input; and

determining whether the incoming email is representative of a security risk based on an output produced by the ML model.

31. The non-transitory computer-readable medium of example 30, the operations further comprising:

performing a remediation action responsive to determining that the incoming email is representative of a security risk.

32. The non-transitory computer-readable medium of example 26, wherein the one or more attributes include a primary attribute and a secondary attribute.

33. The non-transitory computer-readable medium of example 32, wherein said deriving comprises:

extracting the primary attribute from the incoming email; and

determining the secondary attribute based on the primary attribute and additional information associated with the customer.

34. A computer-implemented method comprising:

receiving input indicative of an approval to access emails delivered to an employee of an enterprise;

acquiring an incoming email addressed to the employee;

extracting a primary attribute from the incoming email by parsing content of the incoming email and/or metadata associated with the incoming email;

deriving a secondary attribute based on the primary attribute; and

determining whether the incoming email deviates from past email activity by comparing the primary and secondary attributes to a communication profile associated with the employee.

35. The computer-implemented method of example 34, further comprising:

establishing, via an application programming interface, a connection with an email system employed by the enterprise.

36. The computer-implemented method of example 34, wherein the communication profile includes primary and secondary attributes of past emails delivered to the employee determined to be representative of safe communications.

37. The computer-implemented method of example 36, wherein said determining comprises:

discovering that the primary attribute, the secondary attribute, or the combination of the primary and secondary attributes is not included in the communication profile.

38. The computer-implemented method of example 34, wherein the primary attribute is sender display name, sender username, Sender Policy Framework (SPF) status, DomainKeys Identified Mail (DKIM) status, number of attachments, number of links in a body of the incoming email, a country of origin, information in a header of the incoming email, or an identifier embedded in metadata associated with the incoming email.

39. The computer-implemented method of example 37, further comprising:

establishing that the incoming email does not represent a security risk; and

updating the communication profile by creating an entry that programmatically associates the first and second attributes.

40. A computer-implemented method comprising:

determining, by a threat detection platform, that a communication event involving transmittal of an email is presently occurring;

acquiring, by the threat detection platform, information related to the email;

resolving, by the threat detection platform, entities involved in the communication event by examining the information; and

compiling, by the threat detection platform, corpus statistics for the entities determined to be involved in the communication event.

41. The computer-implemented method of example 40, wherein said determining is accomplished by examining incoming emails received by an email system with which the threat detection platform is programmatically integrated.

42. The computer-implemented method of example 41, wherein the programmatic integration of the threat detection platform with the email system ensures that all external and internal emails are routed through the threat detection platform for examination.

43. The computer-implemented method of example 40, wherein the information is derived from the email.

44. The computer-implemented method of example 40, further comprising:

augmenting, by the threat detection platform, the information with human-curated datasets;

wherein said resolving is performed on the augmented information.

45. The computer-implemented method of example 40, wherein said resolving comprises:

establishing an identity of a sender based on an origin of the incoming email, content of the incoming email, or metadata accompanying the incoming email; and

establishing an identity of a recipient based on a destination of the incoming email, content of the incoming email, or metadata accompanying the incoming email

46. The computer-implemented method of example 40, further comprising:

causing the corpus statistics to be shown in the form of an entity risk graph.

47. The computer-implemented method of example 46, wherein the entities include a sender of the email, a recipient of the email, a domain discovered in the email, a link discovered in the email, an Internet Protocol (IP) address discovered in metadata accompanying the email, an origin of the email, a topic determined based on content of the email, or any combination thereof.

48. The computer-implemented method of example 46, wherein the entity risk graph includes historical combinations of the entities and a separate risk score for each historical combination.

49. The computer-implemented method of example 46, wherein each entity is represented as a separate node in the entity risk graph, and wherein each connection between a pair of nodes is indicative of risk of an event involving a pair of entities associated with the pair of nodes based on past communication events.

50. A non-transitory computer-readable medium with instructions stored thereon that, when executed by a processor, cause the processor to perform operations comprising:

acquiring an incoming email addressed to an employee of an enterprise;

extracting features for the incoming email in the form of primary and secondary attributes;

employing a machine learning (ML) model that consumes the extracted features to determine whether any indicators of compromise that are representative of security threats exist;

generating a signature for each indicator of compromise; and

causing each signature to be ingested by a database to be used in discovering future attacks with the same features.

51. The non-transitory computer-readable medium of example 50, wherein each indicator of compromise is an Internet Protocol (IP) address, email address, Uniform Resource Locator (URL), or domain.

52. The non-transitory computer-readable medium of example 50, the operations further comprising:

performing deep feature extraction to lessen a likelihood of harm from sophisticated security threats.

53. The non-transitory computer-readable medium of example 52, wherein said performing comprises:

applying a deep learning model to understand content, sentiment, and/or tone of the incoming email.

54. The non-transitory computer-readable medium of example 52, wherein said performing comprises:

accessing a landing page by interacting with a link embedded in the incoming email; and

employing a computer vision algorithm to compare the landing page to a set of known sign-on pages verified as authentic.

55. The non-transitory computer-readable medium of example 52, wherein said performing comprises:

employing a crawling algorithm to extract information regarding a secondary link that is embedded in an attachment to the incoming email or accessible via a website linked to by a primary link in the incoming email.

56. A computer-implemented method comprising:

obtaining first data associated with a first batch of past emails received by employees of an enterprise;

generating a first batch of events that is representative of the first batch of past emails;

obtaining second data associated with a second batch of past emails labeled by one or more administrators,

wherein each past email in the second batch of past emails is associated with a label that specifies risk posed to the enterprise;

generating a second batch of events that is representative of the second batch of past emails; and

storing the first and second batches of events in a database.

57. The computer-implemented method of example 56, wherein said generating comprises:

converting the first data associated with each past email in the first batch of past emails into a predefined schema that defines an event.

58. The computer-implemented method of example 56, further comprising:

receiving input indicative of a query for events having a given attribute; and

examining the database to identify events, if any, that have the given attribute.

59. The computer-implemented method of example 58, further comprising:

establishing a count of the identified events; and

causing display of the count on an interface through which the query was submitted.

60. The computer-implemented method of example 57, further comprising:

calculating a risk metric for each past email in the first batch of past emails; and

appending the risk metric calculated for each past email in the first batch of past emails to the corresponding predefined schema.

61. The computer-implemented method of example 60, further comprising:

receiving input indicative of a query for events determined not to represent a threat to security of the enterprise; and

examining the database to identify events, if any, determined not to represent a threat to security of the enterprise; and

causing display of the identified events on an interface through which the query was submitted.

62. The computer-implemented method of example 60, wherein said examining comprises:

parsing the database to determine whether any past emails in the first batch of past emails are associated with a risk metric that falls beneath a threshold; and

parsing the database to determine whether any past emails in the second batch of past emails are associated with a label that indicates no risk is posed.

63. The computer-implemented method of example 56, further comprising:

acquiring an incoming email addressed to an employee of the enterprise;

parsing the incoming email to identity an attribute of the email;

examining the database to identify events, if any, that have the attribute; and

estimating risk posed by the incoming email based on the identified events.

64. A computer-implemented method comprising:

acquiring a series of emails addressed to employees of an enterprise;

identifying entitles involved in the series of emails by examining each email;

creating a series of signatures for the series of emails,

wherein each signature in the series of signatures is associated with a separate email in the series of emails, and

wherein each signature identifies one or more entities determined to be involved in the corresponding email;

acquiring corpus statistics for the entities determined to be involved in the series of emails;

indexing the corpus statistics by date; and

storing the series of signatures and the indexed corpus statistics in a date-partitioned data structure.

65. The computer-implemented method of example 64, further comprising:

acquiring an incoming email addressed to an employee of the enterprise;

identifying at least one entity involved in the incoming email by examining the incoming email; and

comparing the at least one entity against the date-partitioned data structure to establish whether the at least one entity matches any of the series of signatures.

66. The computer-implemented method of example 65, further comprising:

determining that the at least one entity matches a signature in the series of signatures; and

estimating risk posed by the incoming email based on the signature.

67. The computer-implemented method of example 66, wherein said estimating comprises:

determining what risk, if any, was posed by the past email corresponding to the signature; and

calculating a risk metric for the incoming email based on the determined risk of the past email.

68. The computer-implemented method of example 65, further comprising:

determining a degree of similarity between the at least one entity and the series of signatures by employing a machine learning (ML)

algorithm that probabilistically compares the at least one entity to each signature in the series of signatures; and

estimating risk posed by the incoming email based on an output produced by the ML algorithm.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A computer-implemented method comprising:

establishing, via an application programming interface, a connection with a storage medium that includes information regarding digital conduct of employees of an enterprise;

wherein the storage medium is managed by an entity that supports an office suite that is utilized by the employees of the enterprise;

downloading, via the application programming interface, a first series of past communications received by an employee over a first interval of time into a local processing environment;

building a machine learning (ML) model for the employee by providing the first series of past communications to the ML model as training data, so as to train the ML model to understand normal traits and content of communications received by the employee;

receiving, via the application programming interface, a communication addressed to the employee; and

establishing whether the communication represents a security risk by applying the ML model to the communication.

2. The computer-implemented method of claim 1, further comprising:

receiving input indicative of an approval from an administrator associated with the enterprise to access the information in the storage medium;

wherein said establishing is performed in response to receiving the input.

3. The computer-implemented method of claim 1, wherein the first series of past communications includes multiple emails that were delivered to the employee.

4. The computer-implemented method of claim 1, further comprising:

examining each past communication in the first series of past communications to establish attributes of the first series of past communications; and

providing the attributes derived from the first series of past communications to the ML model as training data.

5. The computer-implemented method of claim 1, further comprising:

determining, based on an output produced by the ML model, that the communication represents a security risk; and

characterizing the security risk along multiple dimensions.

6. The computer-implemented method of claim 5, wherein the multiple dimensions include:

an attacked party,

an attack vector,

an impersonated party,

an impersonation strategy, and

an attack goal.

7. The computer-implemented method of claim 1, wherein the first series of past communications includes all emails received by the employee during the first interval of time.

8. The computer-implemented method of claim 1, further comprising:

downloading, via the application programming interface, a second series of past communications corresponding to a second interval of time that precedes the first interval of time into the local processing environment; and

establishing whether any communications received during the second interval of time represent security risks by applying the ML model to the second series of past communications.

9. The computer-implemented method of claim 1, further comprising:

examining the communication to establish multiple attributes;

generating a statistical profile for the communication,

wherein the statistical profile includes a risk score for each pair of attributes included in the multiple attributes, each risk score being based on riskiness of historical communications involving the corresponding pair of attributes.

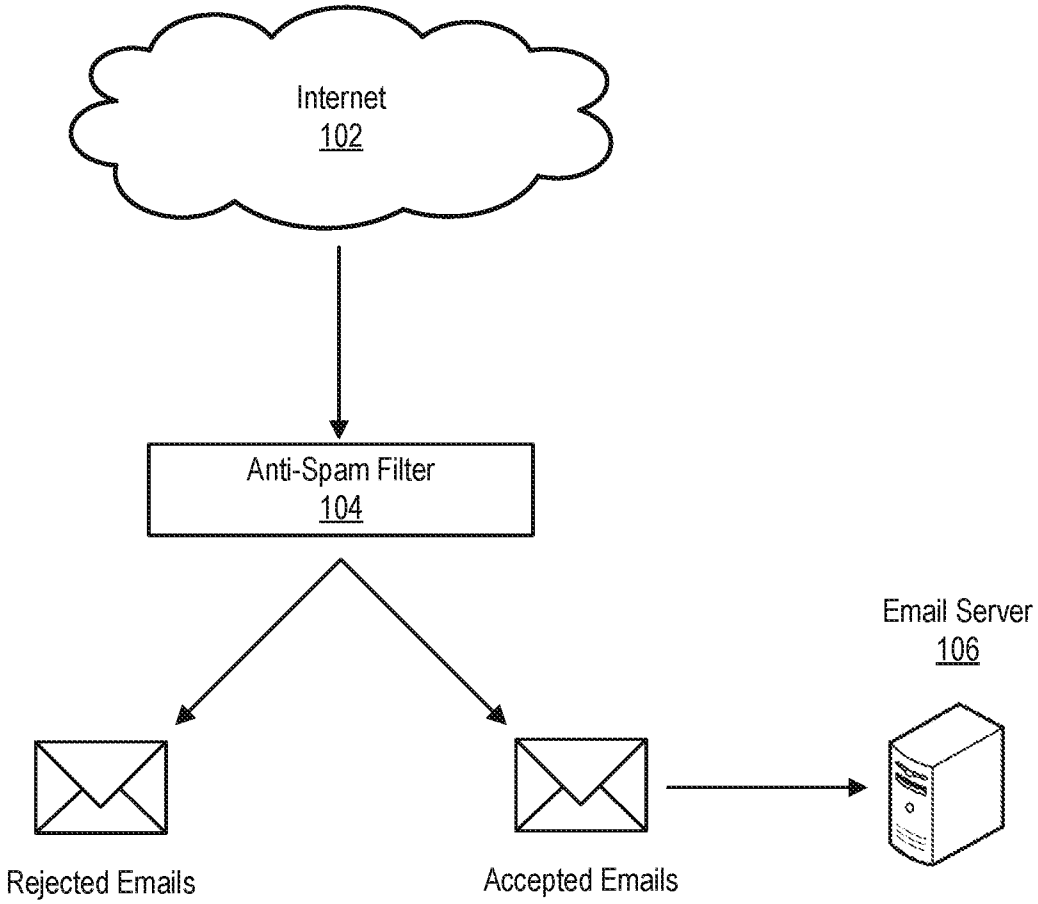
Internet
102
Anti-Spam Filter
104
Email Server
106
Rejected Emails
Accepted Emails

FIG. 1

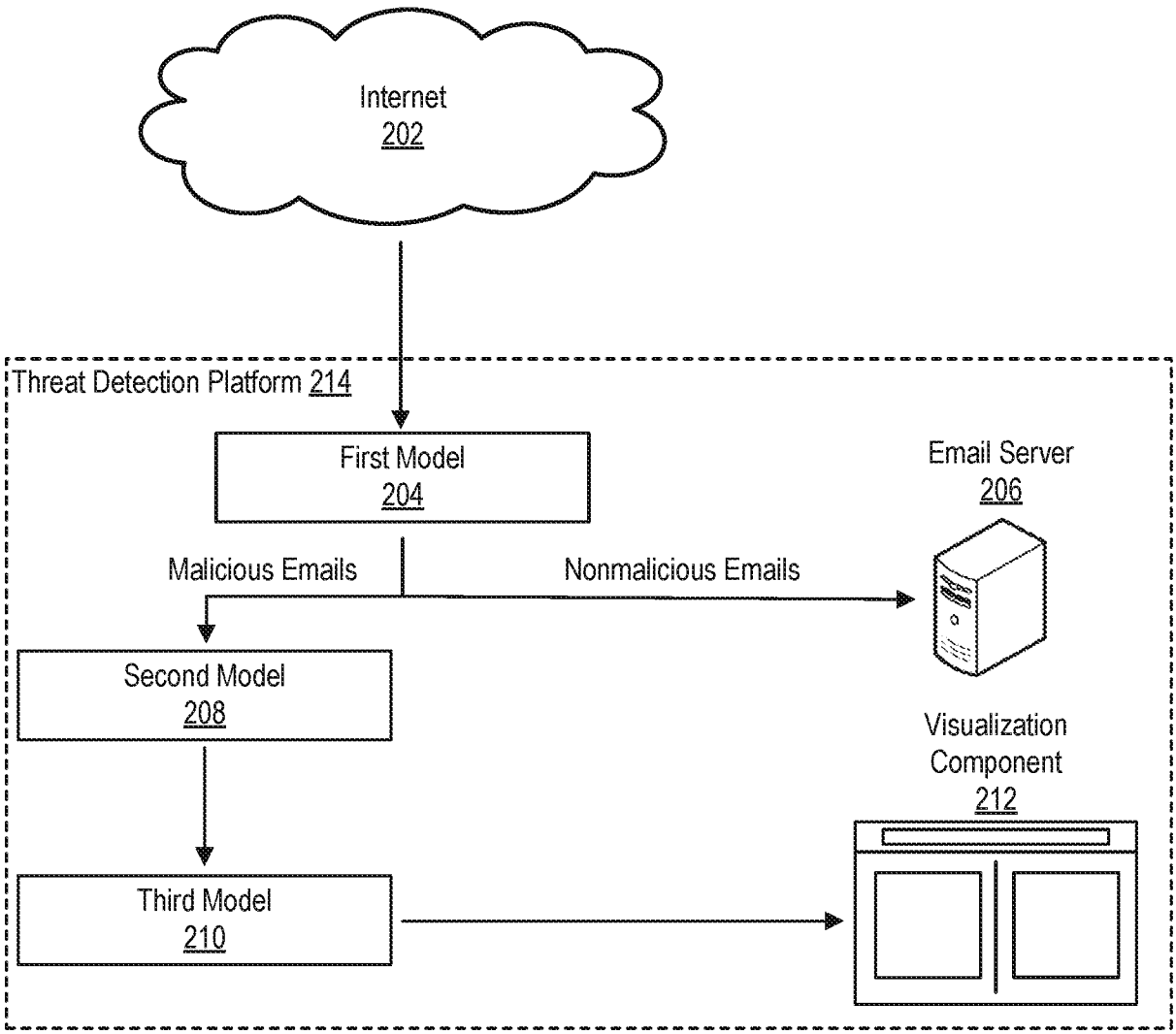
Internet
202
Threat Detection Platform 214
First Model
204
Email Server
206
Malicious Emails
Nonmalicious Emails
Second Model
208
Visualization
Component
212
Third Model
210

FIG. 2

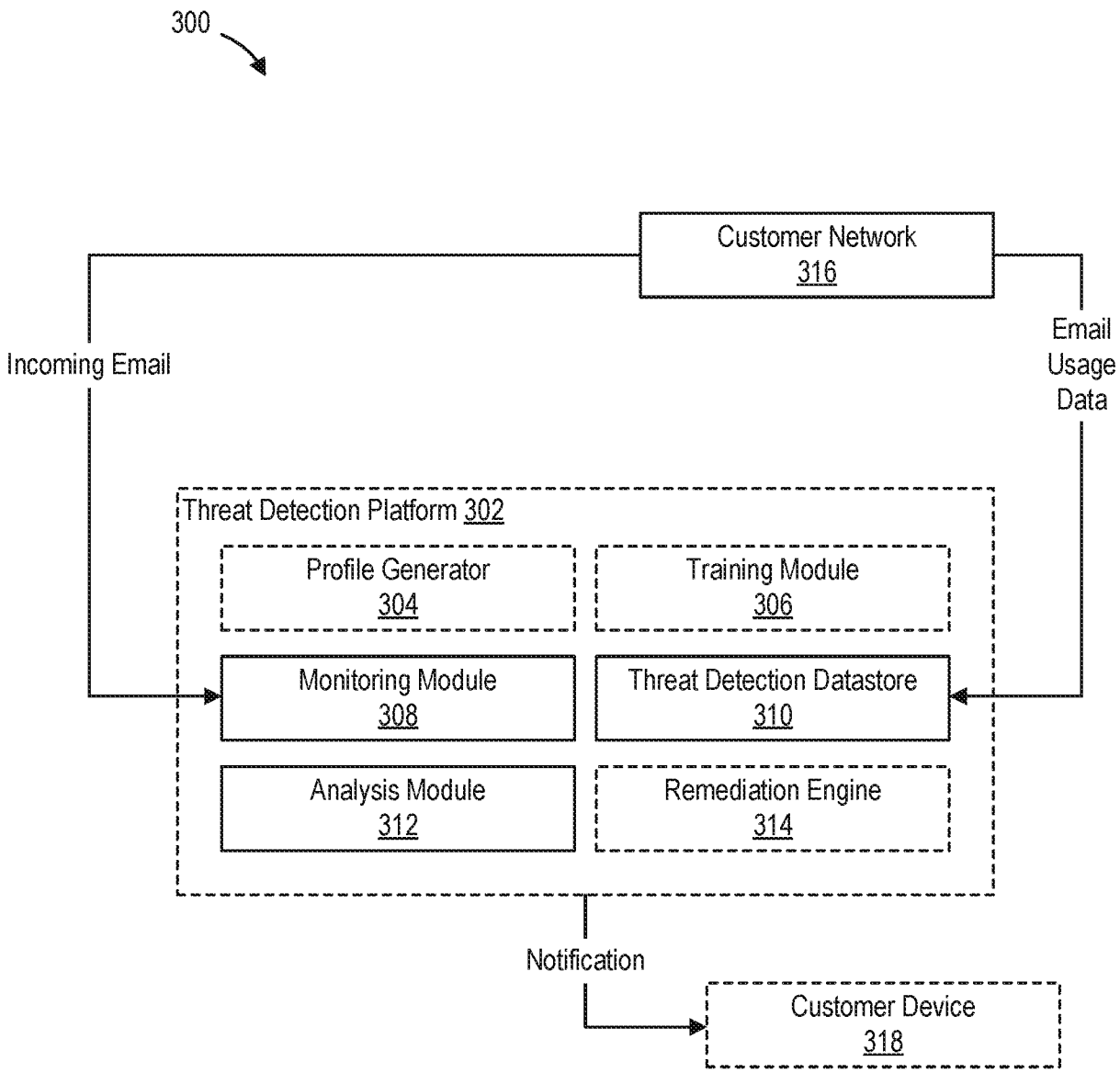
300
Customer Network
316
Incoming Email
Email
Usage
Data
Threat Detection Platform 302
Profile Generator
304
Training Module
306
Monitoring Module
308
Threat Detection Datastore
310
Analysis Module
312
Remediation Engine
314
Notification
Customer Device
318

FIG. 3

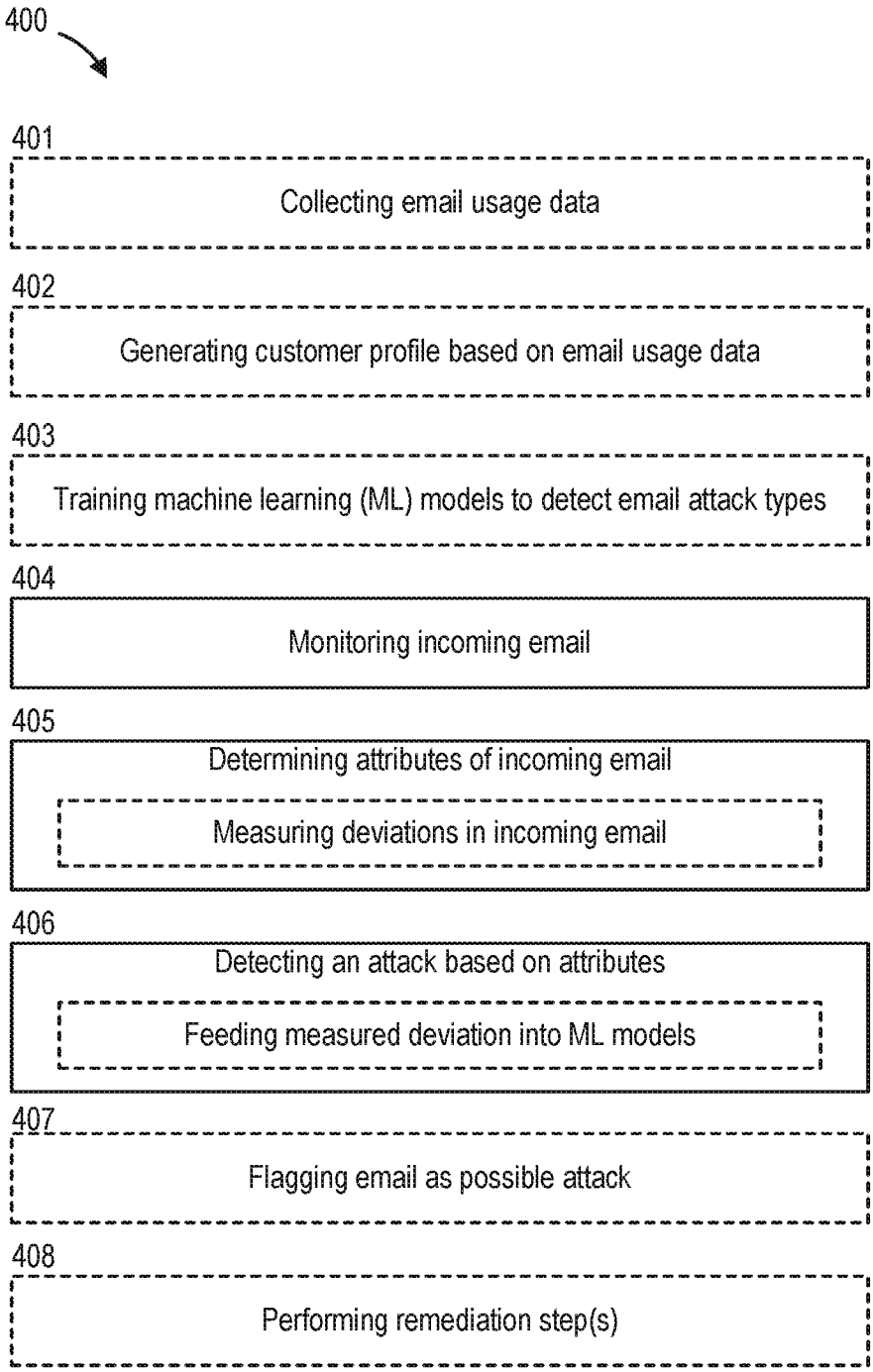
400
401
Collecting email usage data
402
Generating customer profile based on email usage data
403
Training machine learning (ML) models to detect email attack types
404
Monitoring incoming email
405
Determining attributes of incoming email
Measuring deviations in incoming email
406
Detecting an attack based on attributes
Feeding measured deviation into ML models
407
Flagging email as possible attack
408
Performing remediation step(s)

FIG. 4

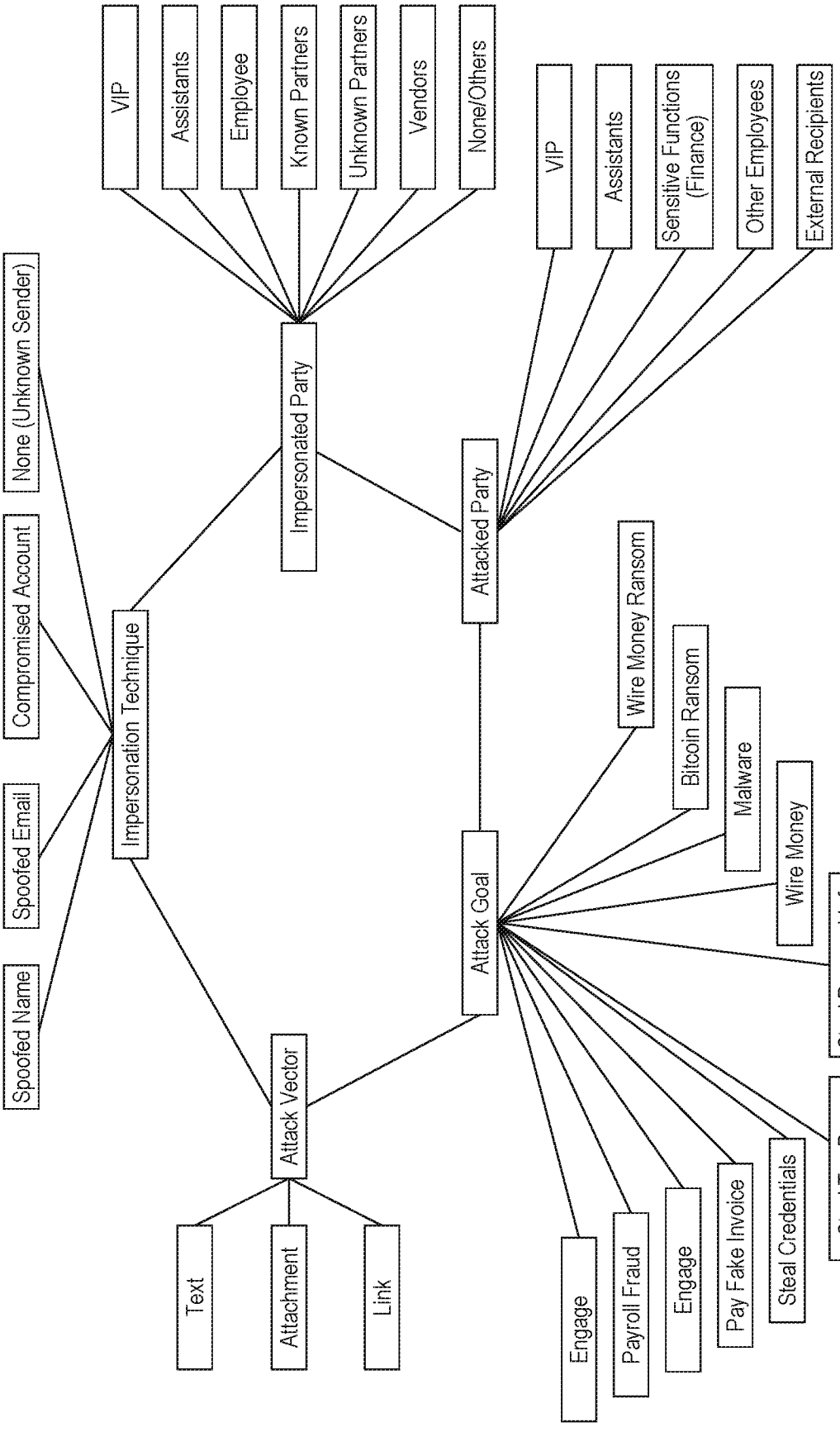
Impersonation Technique
Spoofed Name
Spoofed Email
Compromised Account
None (Unknown Sender)
Impersonated Party
VIP
Assistants
Employee
Known Partners
Unknown Partners
Vendors
None/Others
Attacked Party
VIP
Assistants
Sensitive Functions (Finance)
Other Employees
External Recipients
Attack Vector
Text
Attachment
Link
Attack Goal
Engage
Payroll Fraud
Engage
Pay Fake Invoice
Steal Credentials
Steal Tax Docs
Steal Personal Info
Wire Money
Malware
Bitcoin Ransom
Wire Money Ransom

FIG. 5

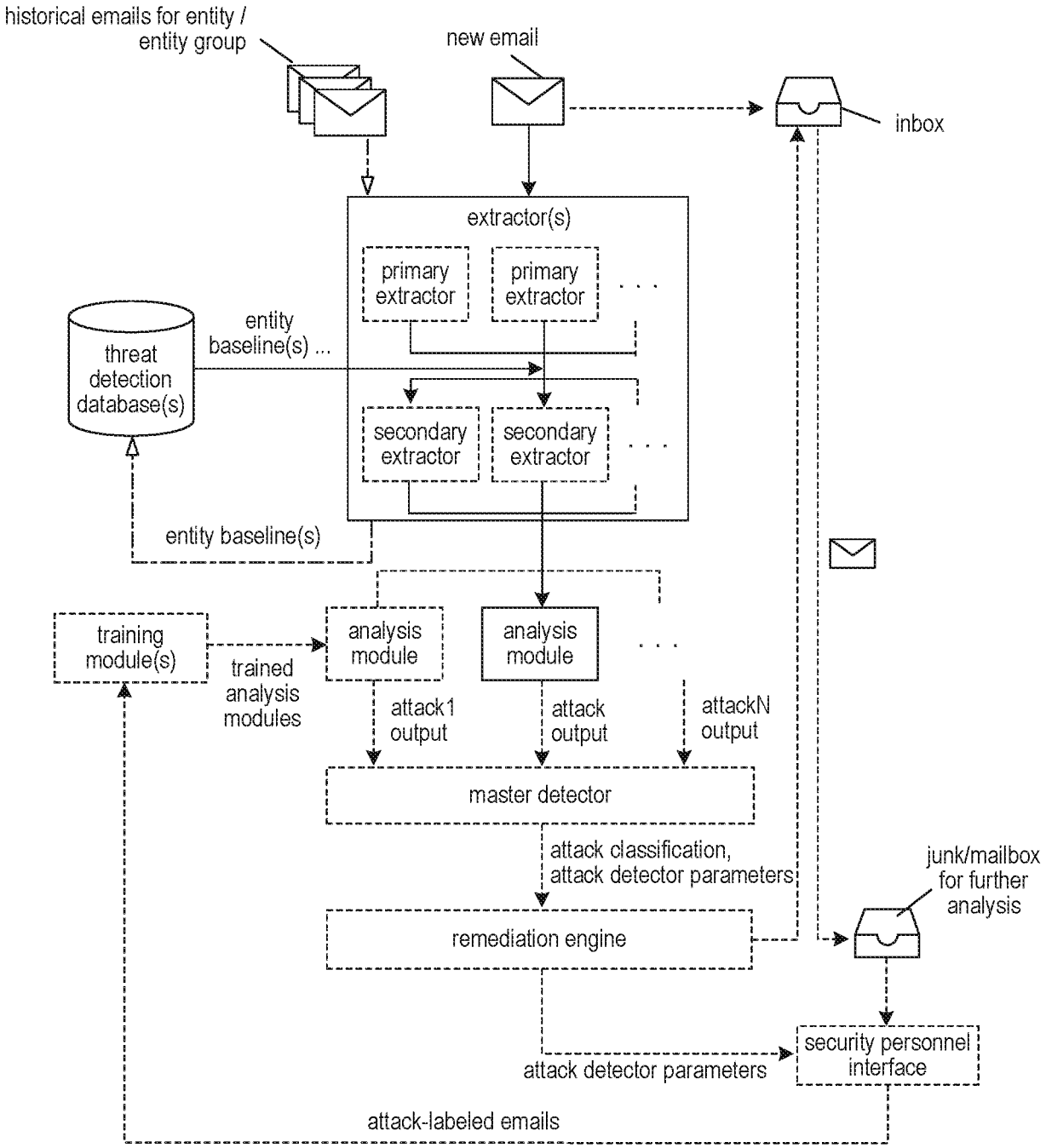
historical emails for entity / entity group
new email
inbox
extractor(s)
primary extractor
primary extractor
secondary extractor
secondary extractor
threat detection database(s)
entity baseline(s) ...
entity baseline(s)
training module(s)
trained analysis modules
analysis module
analysis module
attack1 output
attack output
attackN output
master detector
attack classification, attack detector parameters
remediation engine
junk/mailbox for further analysis
security personnel interface
attack detector parameters
attack-labeled emails

FIG. 6

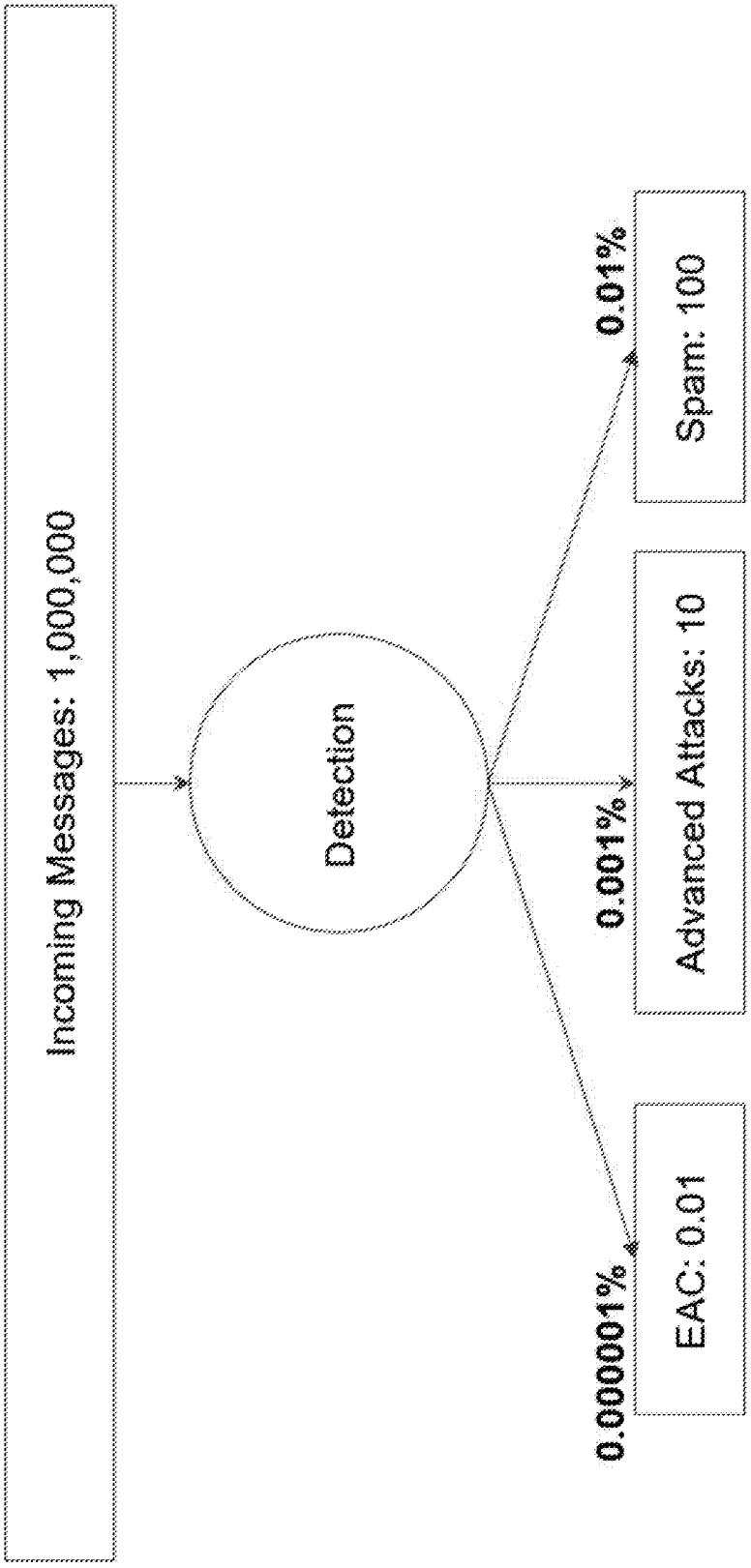
Incoming Messages: 1,000,000
Detection
0.01%
Spam: 100
0.001%
Advanced Attacks: 10
0.000001%
EAC: 0.01

FIG. 7

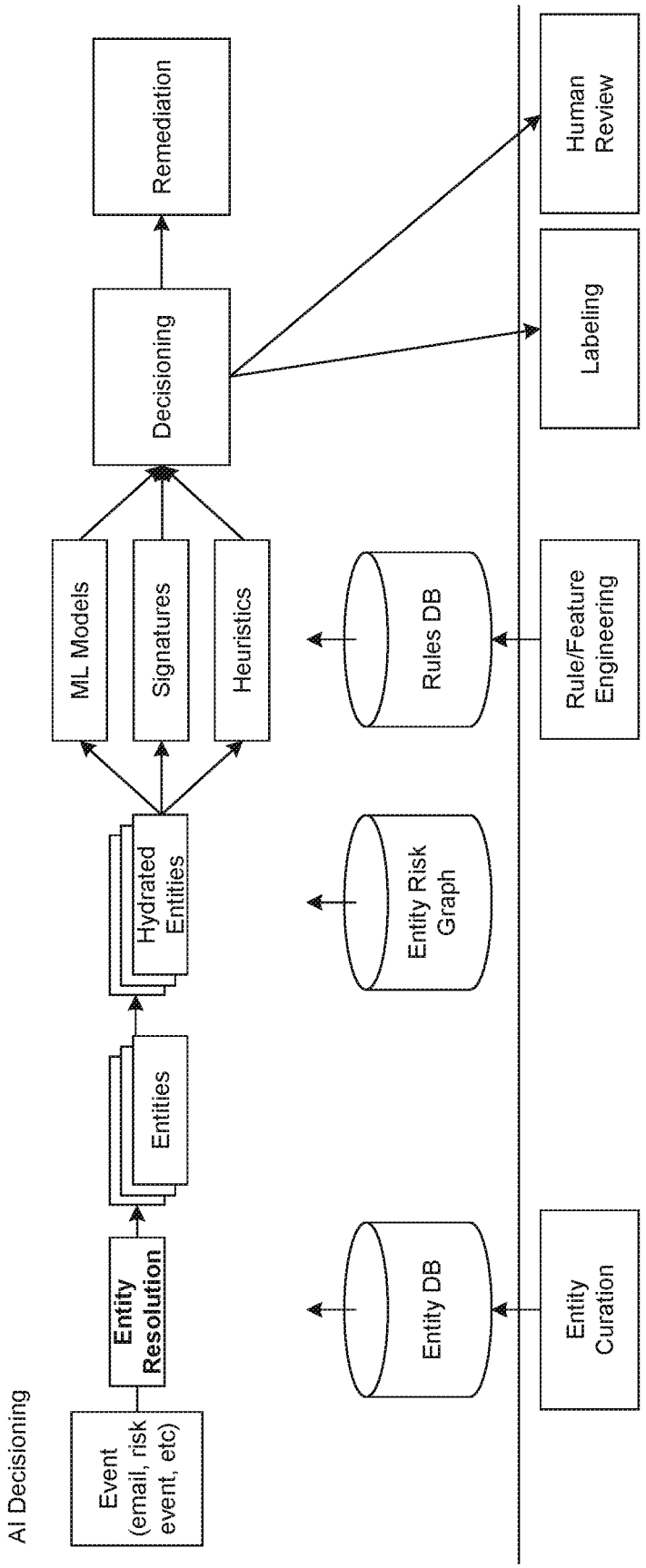
AI Decisioning
Event (email, risk event, etc)
Entity Resolution
Entities
Hydrated Entities
ML Models
Signatures
Heuristics
Decisioning
Remediation
Entity DB
Entity Risk Graph
Rules DB
Entity Curation
Rule/Feature Engineering
Labeling
Human Review
Human Computation

FIG. 8A

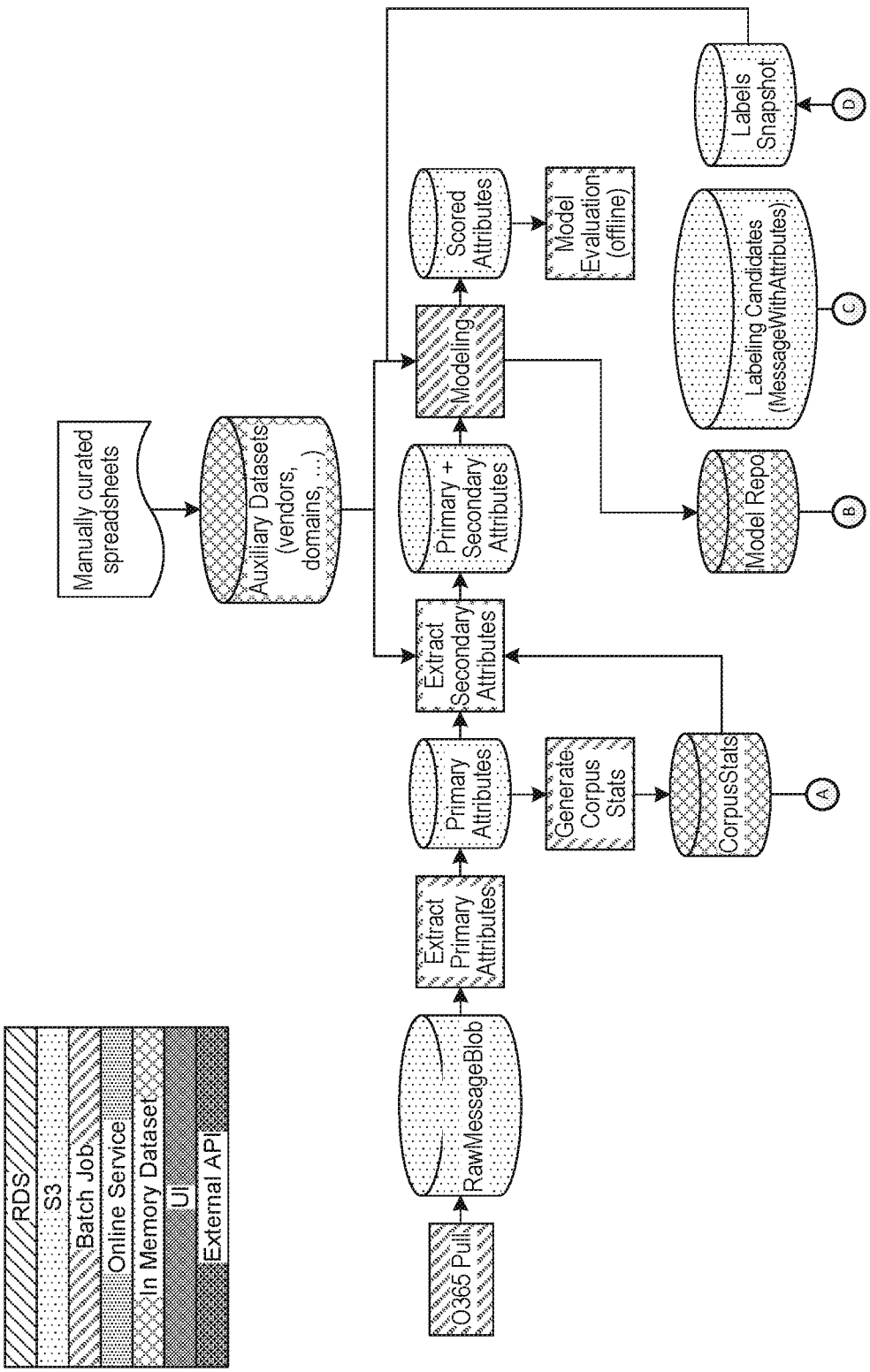
RDS
S3
Batch Job
Online Service
In Memory Dataset
UI
External API
O365 Pull
RawMessageBlob
Extract Primary Attributes
Primary Attributes
Generate Corpus Stats
CorpusStats
A
Extract Secondary Attributes
Primary + Secondary Attributes
Modeling
Scored Attributes
Model Evaluation (offline)
Manually curated spreadsheets
Auxiliary Datasets (vendors, domains, ...)
Model Repo
B
Labeling Candidates (MessageWithAttributes)
C
Labels Snapshot
D

FIG. 8B

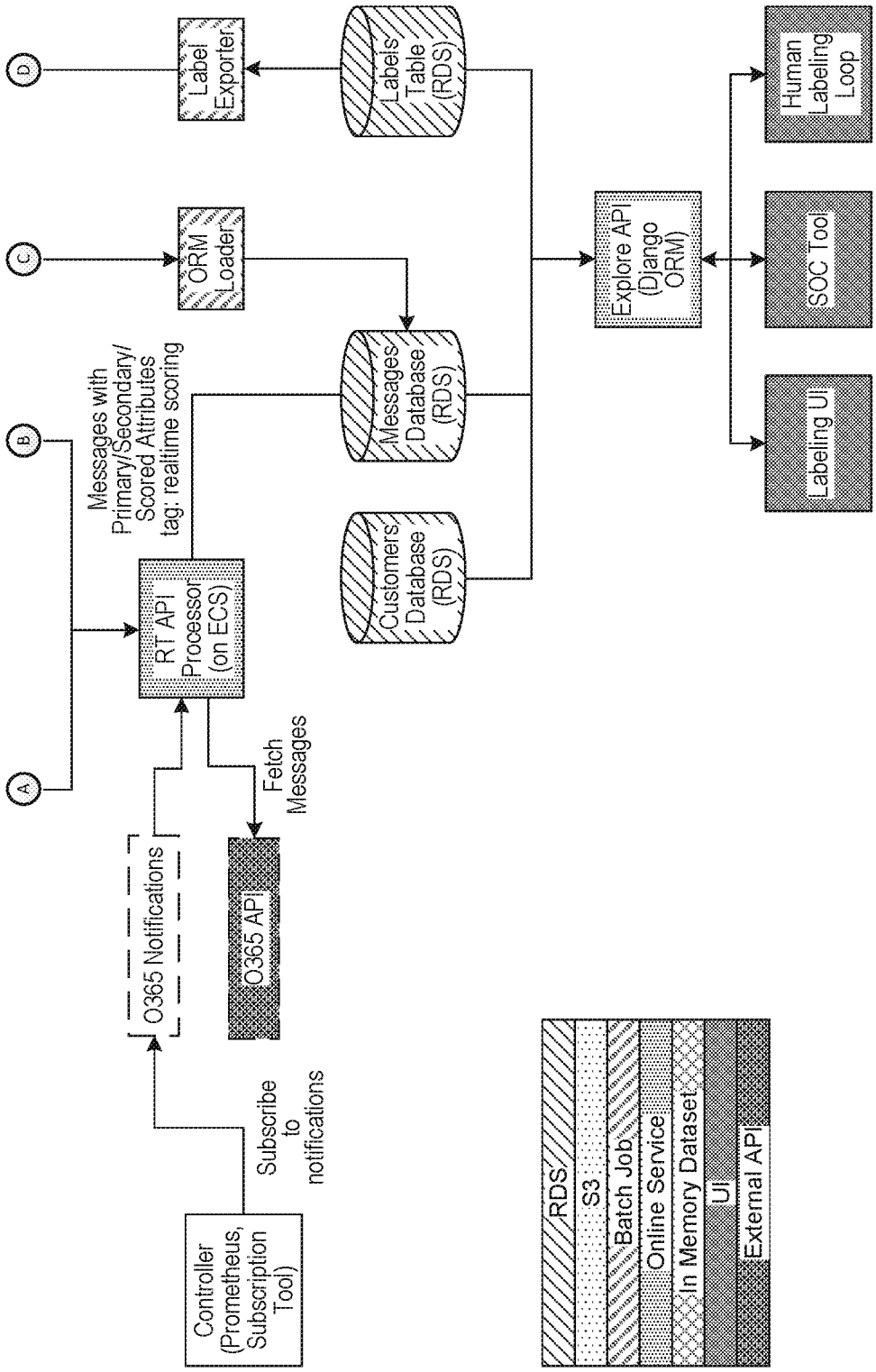
A
B
C
D
Controller (Prometheus, Subscription Tool)
Subscribe to notifications
O365 Notifications
O365 API
Fetch Messages
RT API Processor (on ECS)
Messages with Primary/Secondary/ Scored Attributes tag: realtime scoring
ORM Loader
Label Exporter
Messages Database (RDS)
Customers Database (RDS)
Labels Table (RDS)
Explore API (Django ORM)
Labeling UI
SOC Tool
Human Labeling Loop
RDS
S3
Batch Job
Online Service
In Memory Dataset
UI
External API

FIG. 8B, Cont.

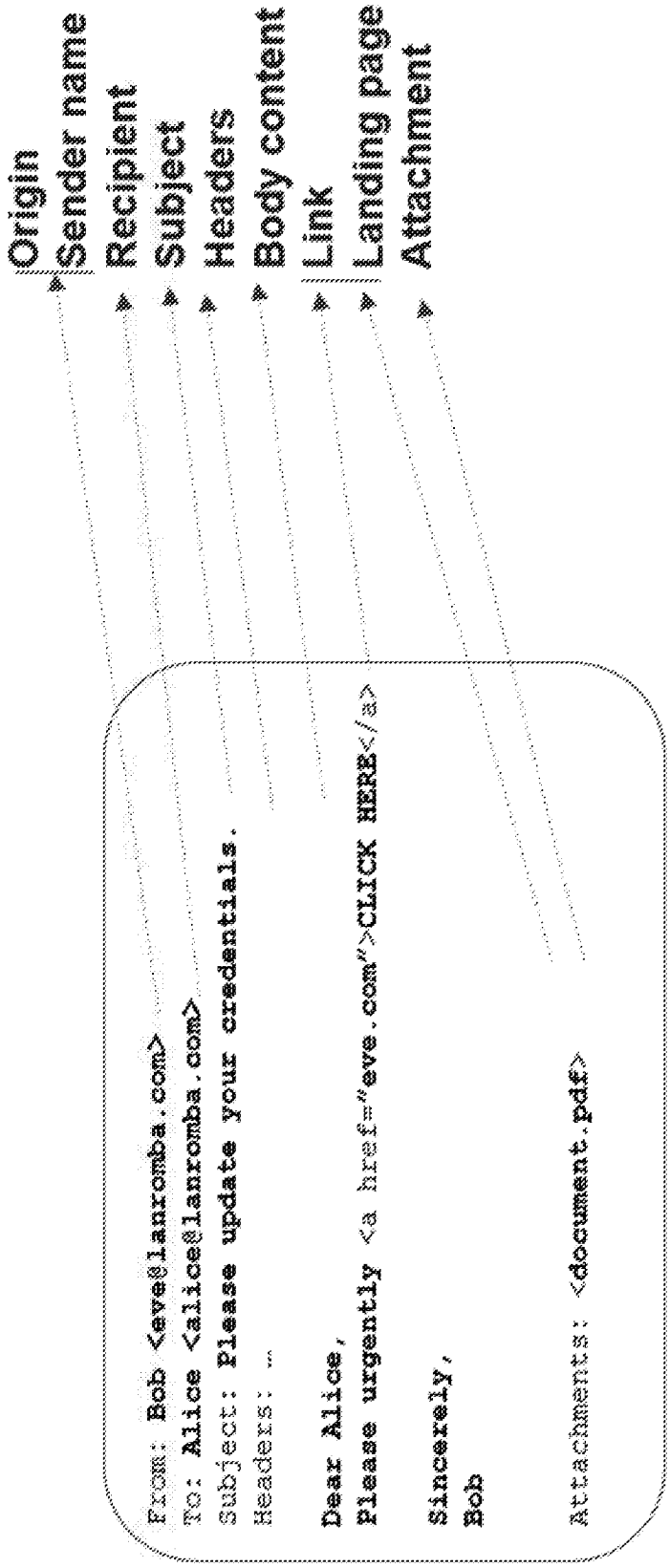
Origin
Sender name
Recipient
Subject
Headers
Body content
Link
Landing page
Attachment
From: Bob <eve@lanromba.com>
To: Alice <alice@lanromba.com>
Subject: Please update your credentials.
Headers: …
Dear Alice,
Please urgently CLICK HERE
Sincerely,
Bob
Attachments: <document.pdf>

FIG. 9

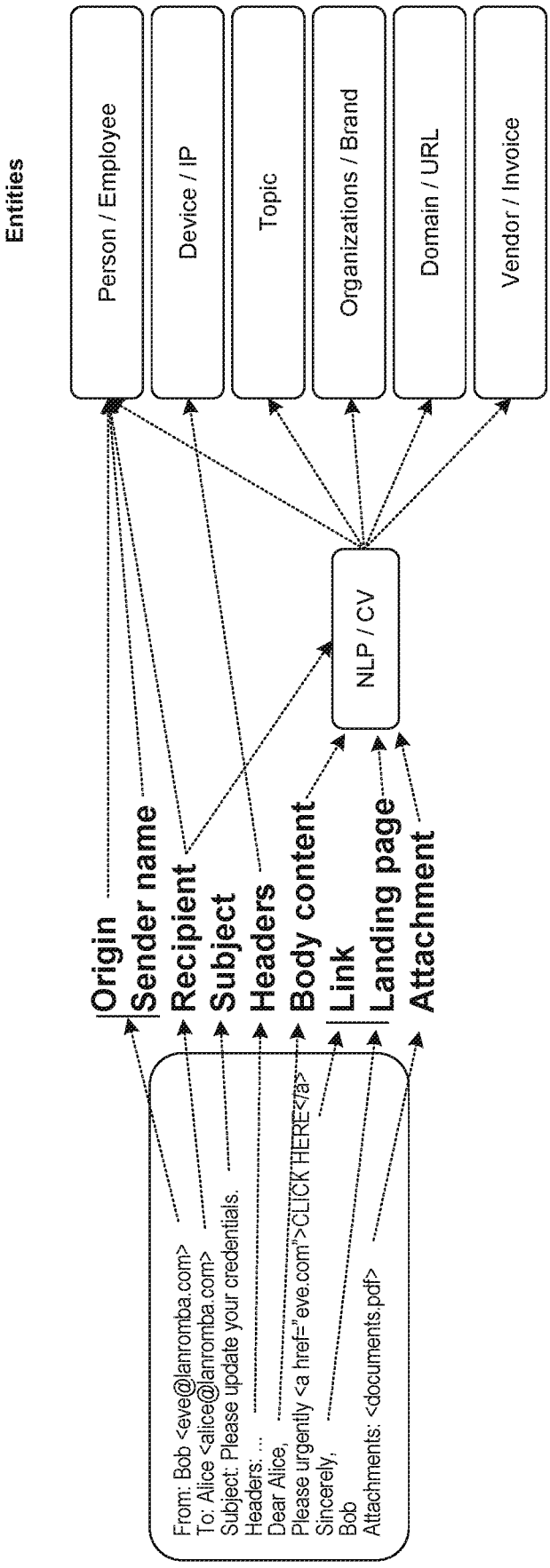
Entities
Person / Employee
Device / IP
Topic
Organizations / Brand
Domain / URL
Vendor / Invoice
NLP / CV
Origin
Sender name
Recipient
Subject
Headers
Body content
Link
Landing page
Attachment
From: Bob <eve@lanromba.com>
To: Alice <alice@lanromba.com>
Subject: Please update your credentials.
Headers: ...
Dear Alice,
Please urgently CLICK HERE
Sincerely,
Bob
Attachments: <documents.pdf>

FIG. 10A

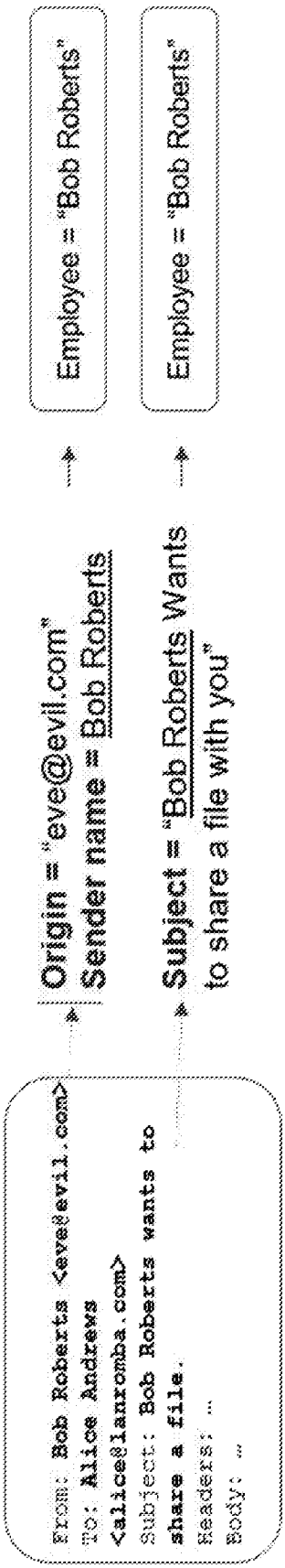
From: Bob Roberts <eve@evil.com>
To: Alice Andrews
<alice@ianromba.com>
Subject: Bob Roberts wants to
share a file.
Headers: ...
Body: ...
Origin = "eve@evil.com"
Sender name = Bob Roberts
Subject = "Bob Roberts Wants
to share a file with you"
Employee = "Bob Roberts"
Employee = "Bob Roberts"

FIG. 10B

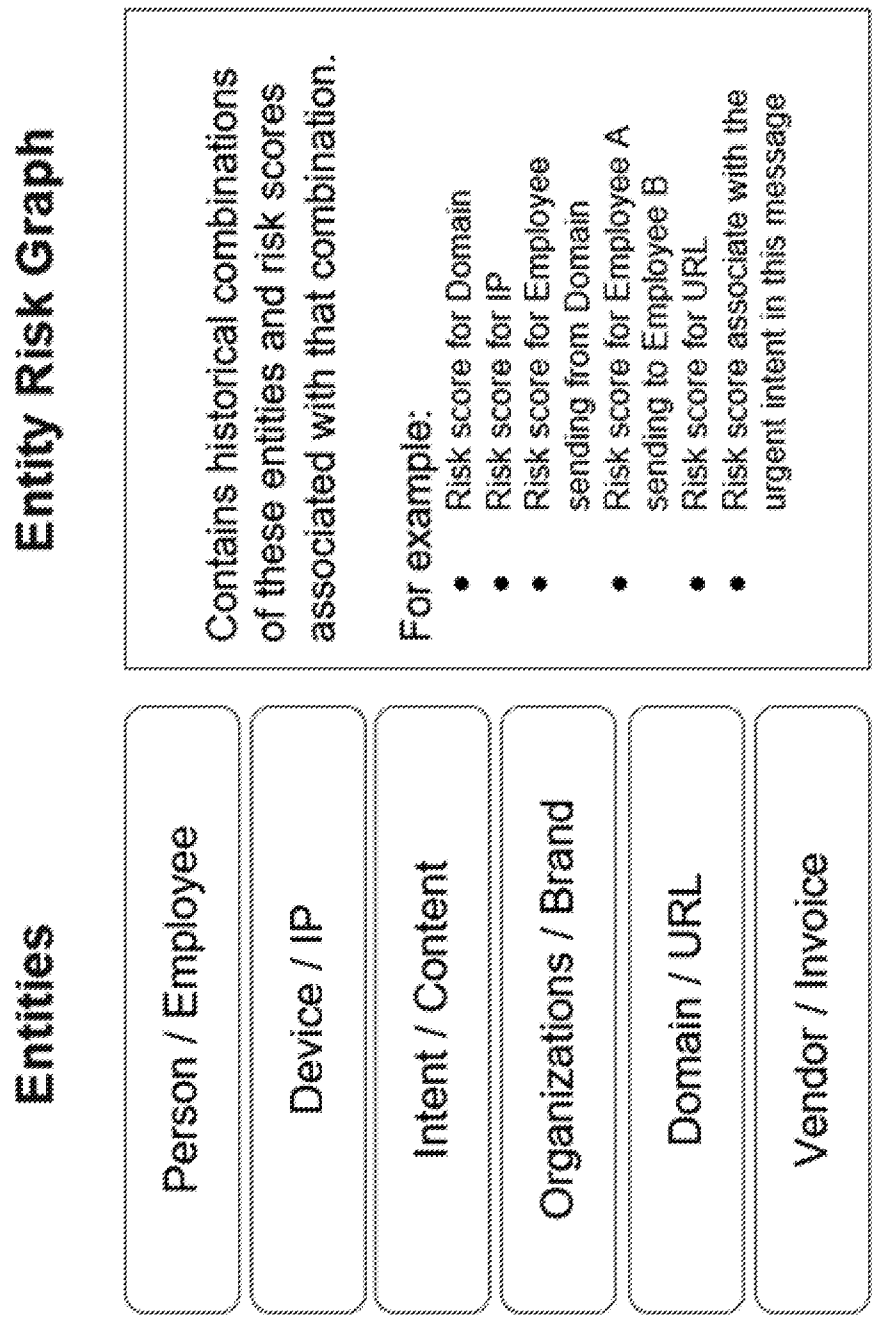
Entities
Person / Employee
Device / IP
Intent / Content
Organizations / Brand
Domain / URL
Vendor / Invoice
Entity Risk Graph
Contains historical combinations of these entities and risk scores associated with that combination.
For example:
Risk score for Domain
Risk score for IP
Risk score for Employee sending from Domain
Risk score for Employee A sending to Employee B
Risk score for URL
Risk score associate with the urgent intent in this message

FIG. 11

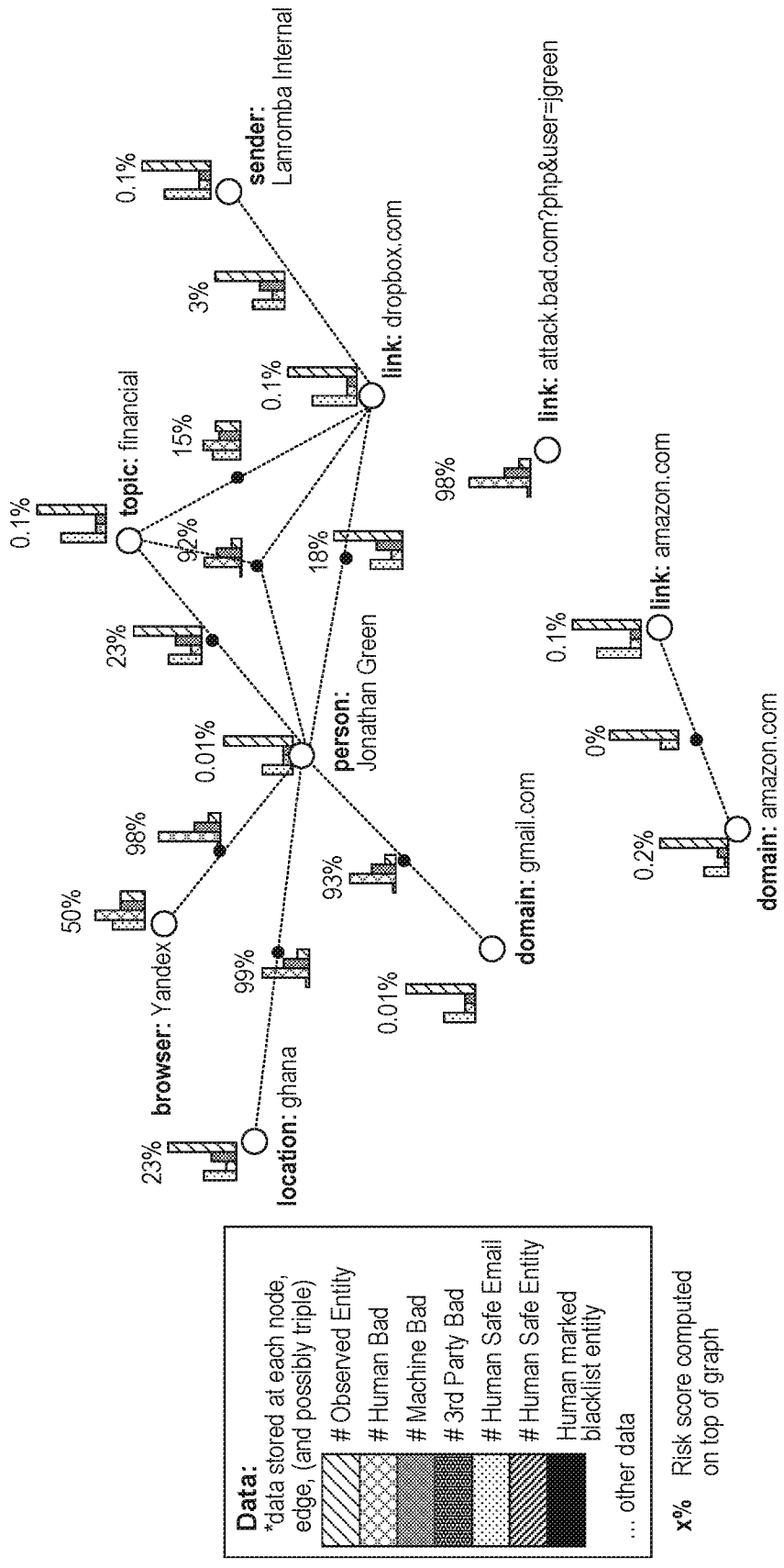
Data:
*data stored at each node, edge, (and possibly triple)
Observed Entity
Human Bad
Machine Bad
3rd Party Bad
Human Safe Email
Human Safe Entity
Human marked blacklist entity
... other data
x% Risk score computed on top of graph
location: ghana
browser: Yandex
person: Jonathan Green
topic: financial
sender: Lanromba Internal
link: dropbox.com
link: attack.bad.com?php&user=jgreen
link: amazon.com
domain: gmail.com
domain: amazon.com
23%
50%
98%
0.01%
99%
93%
0.01%
23%
0.1%
92%
18%
15%
0.1%
3%
0.1%
98%
0.1%
0%
0.2%

FIG. 12

Stage	Output(s)	Data Dependencies
Stage 1: Download	Raw Messages Raw Attachments Raw Outlook Directory List* Raw Mail Filters* * : Output may persisted in memory, or may be temporarily stored in memory before being fed into Stage 2.	None
Stage 2: Extraction Threat Detection Platform (TDP) can extract text, metadata, and/or attachment signals from an email message directly, or the TDP may use learned model parameter(s), such as embedding models, for text extraction.	Text Signals Attachment Signals Metadata Signals ("Primary Attributes") Output can be fed into Stage 3 in, for example, columnar format.	Raw Messages Raw Attachments Raw Outlook Directory List Raw Mail Filters Raw Risk Events Model DB (e.g., Text Model, Attachment Model) Rules DB (e.g., for text-extraction rules)
Stage 3: Entity Resolution	Canonical identifiers of entities (e.g., people brands, vendors)	
Stage 4: Summarization TDP can produce summaries of historical features ("corpus stats").	Key-value DBs (Date, Key) indexed summaries	Text Signals Attachment Signals Metadata Signals Summarization Cfg

FIG. 13

Stage	Output(s)	Data Dependencies
Stage 5: Hydration and Scoring DAG of rules/models that have access to join against auxiliary databases.	Derived Attributes ("Secondary Attributes") Score Attributes (e.g., based on rules that define a likelihood of an attack) Output can be fed into Stage 6 in, for example, columnar format.	Model DB (e.g., Sub-models and Detection Models) Rules DB (e.g., User-defined Feature Rules and Detection Rules) Summary DB ("Corpus Stats") Employee DB Brand DB Domain DB URL DB Signature DB
Stage 6: Final Detection Decision Each score attribute may have a threshold that is updated to maintain a target percentage flagged.	Is attack in {BORDERLINE, SUSPICIOUS, BAD} May be stored in memory only.	Threshold DB

FIG. 13, Cont.

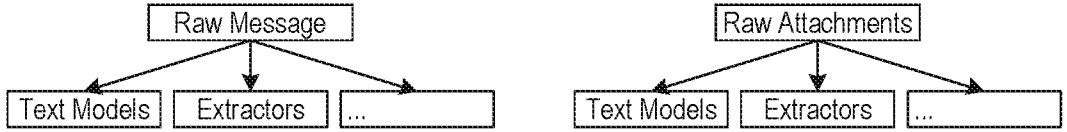

FIG. 14A

Metadata Attributes Table *(currently equivalent to Text Signals + Primary Attributes)*:

Time	MsgID	Sender ID	Mailbox ID	Auth	From FQDN	URLS	phrases	bigrams	trigrams	Tokenized words	Word embeddings	⋮	Text-model prediction	⋮	Image hash	⋮	signature	⋮

FIG. 14B

Attachment Attributes Table *(currently equivalent to Attachment Signals)*:

Time	Attachment ID	URLs	Image Hashes	phrases	bigrams	trigrams	Attachment metadata	⋮	Text-model prediction	...	...

FIG. 14C

Date Key	Summary Type	Summary Key	This Count	Total Count
2019-01-01	Sender Registered Domain	gmail.com	1000	10000000
2019-01-01	Sender Email	Jeremy Zimmer x zimmerj@uta.com	12	15
2019-01-01	Sender Domain x PDF Creator	Westin.com x Free PDFCreator	0	3

FIG. 14D

Date Key	Person ID	Emails Used	Person	Level	...
2019-01-01	ID(Jeremy Zimmer)	100: zimmerj@uta.co, 5: zimmer@gmail.com	Employee of UTA	CEO	...

FIG. 14E

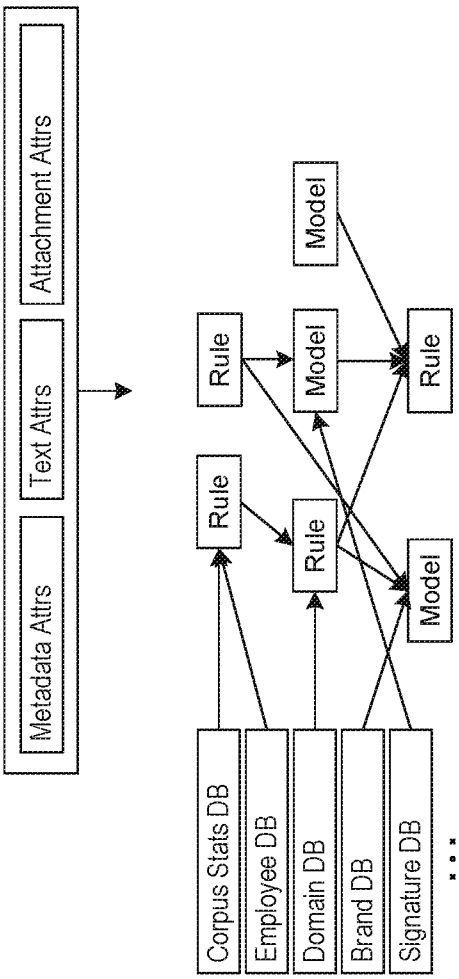

FIG. 14F

FIG. 14G

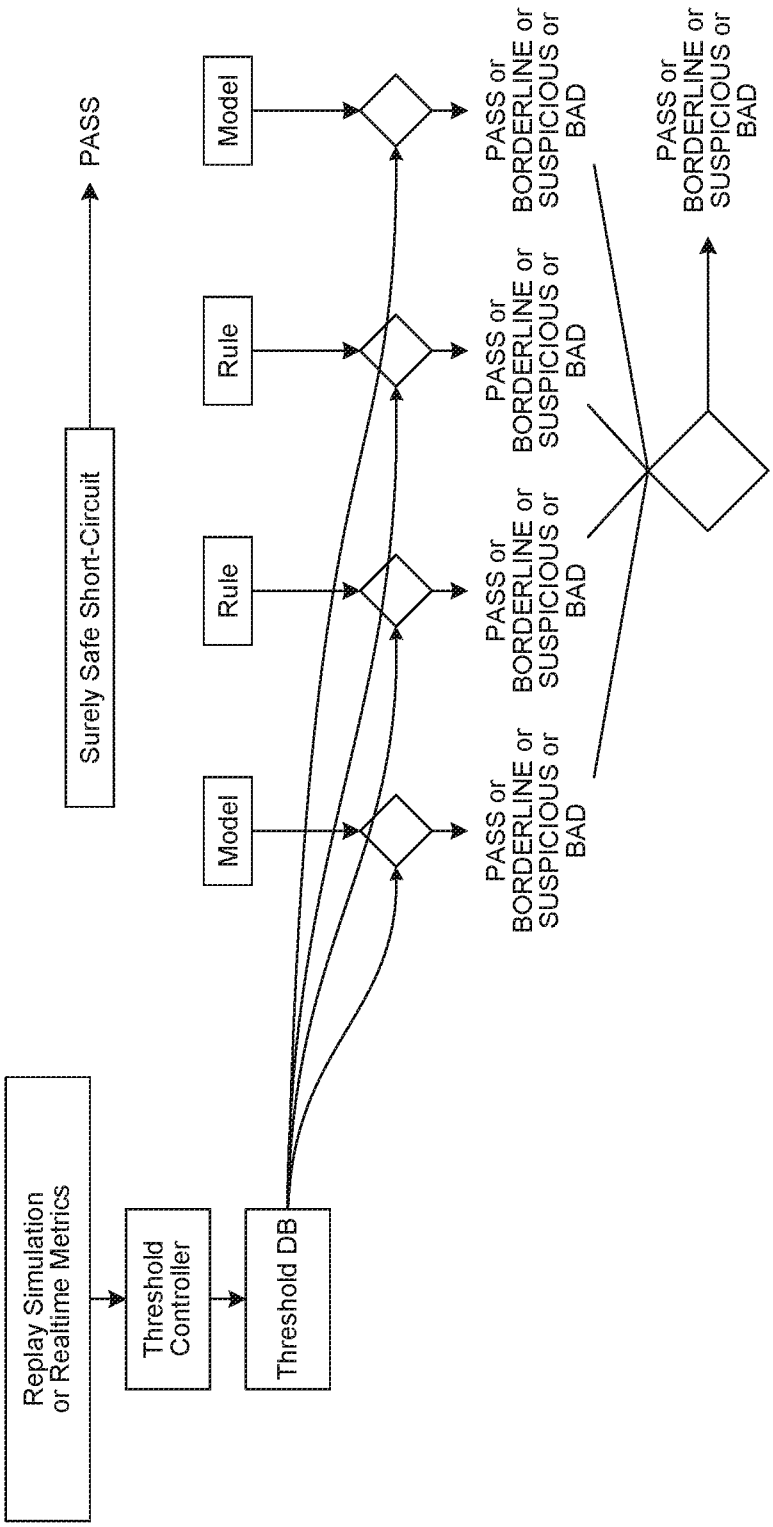
Replay Simulation or Realtime Metrics
Threshold Controller
Threshold DB
Model
Rule
Rule
Model
Surely Safe Short-Circuit
PASS
PASS or BORDERLINE or SUSPICIOUS or BAD
PASS or BORDERLINE or SUSPICIOUS or BAD
PASS or BORDERLINE or SUSPICIOUS or BAD
PASS or BORDERLINE or SUSPICIOUS or BAD

FIG. 14H

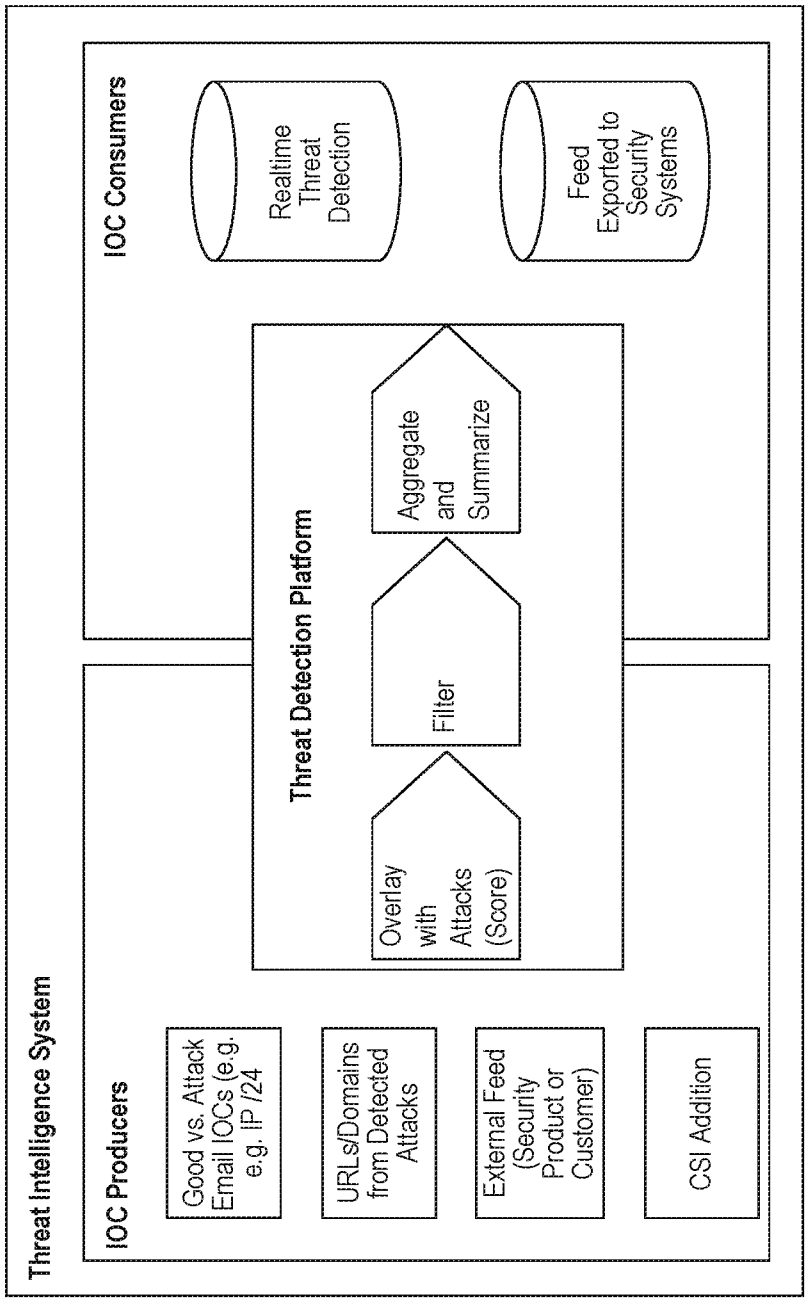
Threat Intelligence System
IOC Producers
Good vs. Attack Email IOCs (e.g. e.g. IP /24
URLs/Domains from Detected Attacks
External Feed (Security Product or Customer)
CSI Addition
Threat Detection Platform
Overlay with Attacks (Score)
Filter
Aggregate and Summarize
IOC Consumers
Realtime Threat Detection
Feed Exported to Security Systems

FIG. 15A

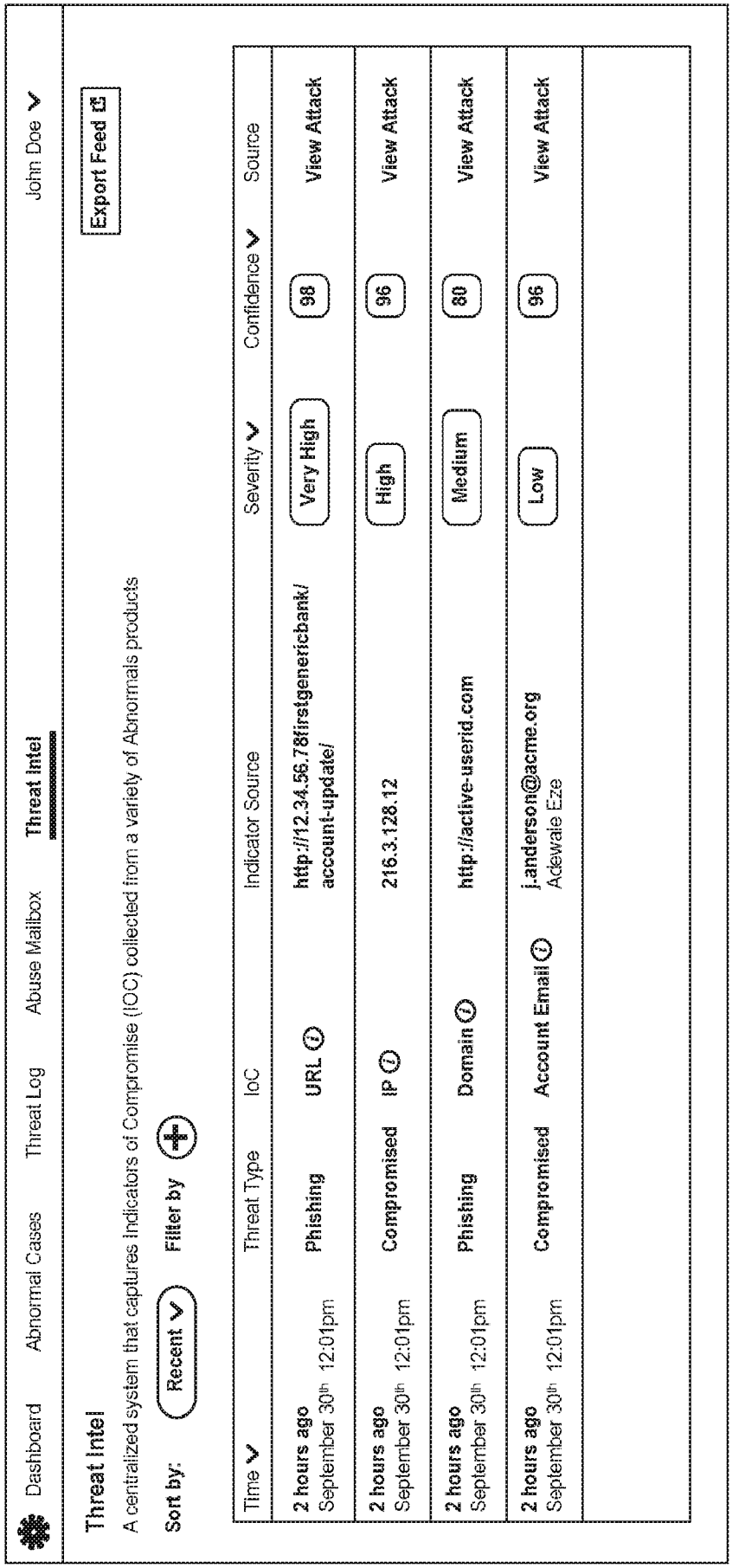
Dashboard
Abnormal Cases
Threat Log
Abuse Mailbox
Threat Intel
John Doe
Threat Intel
A centralized system that captures Indicators of Compromise (IOC) collected from a variety of Abnormals products
Export Feed
Sort by:
Recent
Filter by
Time
Threat Type
IoC
Indicator Source
Severity
Confidence
Source
2 hours ago
September 30th 12:01pm
Phishing
URL
http://12.34.56.78firstgenerichank/account-update/
Very High
98
View Attack
2 hours ago
September 30th 12:01pm
Compromised
IP
216.3.128.12
High
96
View Attack
2 hours ago
September 30th 12:01pm
Phishing
Domain
http://active-userid.com
Medium
80
View Attack
2 hours ago
September 30th 12:01pm
Compromised
Account Email
j.anderson@acme.org
Adewale Eze
Low
96
View Attack

FIG. 15B

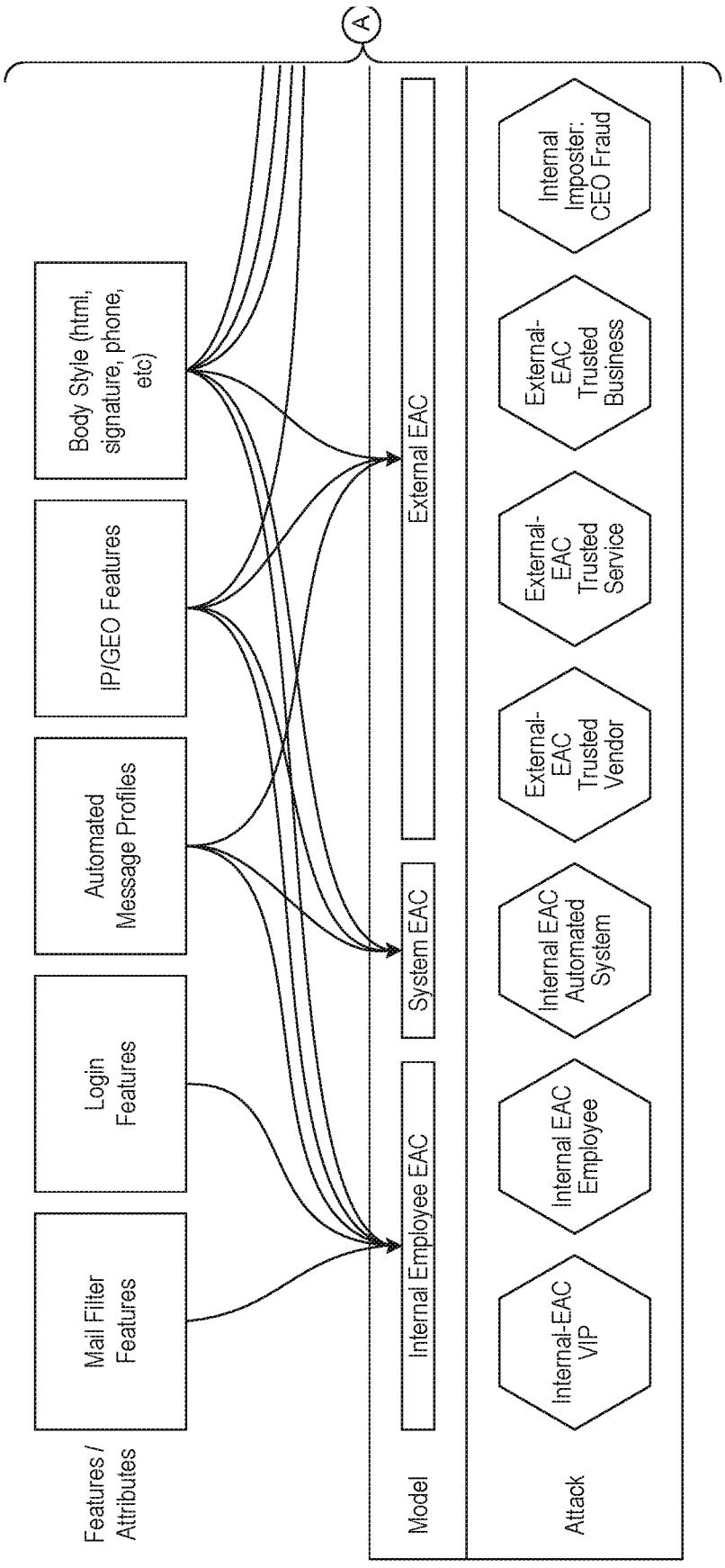
A
Features / Attributes
Mail Filter Features
Login Features
Automated Message Profiles
IP/GEO Features
Body Style (html, signature, phone, etc)
Model
Internal Employee EAC
System EAC
External EAC
Attack
Internal-EAC VIP
Internal EAC Employee
Internal EAC Automated System
External-EAC Trusted Vendor
External-EAC Trusted Service
External-EAC Trusted Business
Internal Imposter: CEO Fraud

FIG. 16

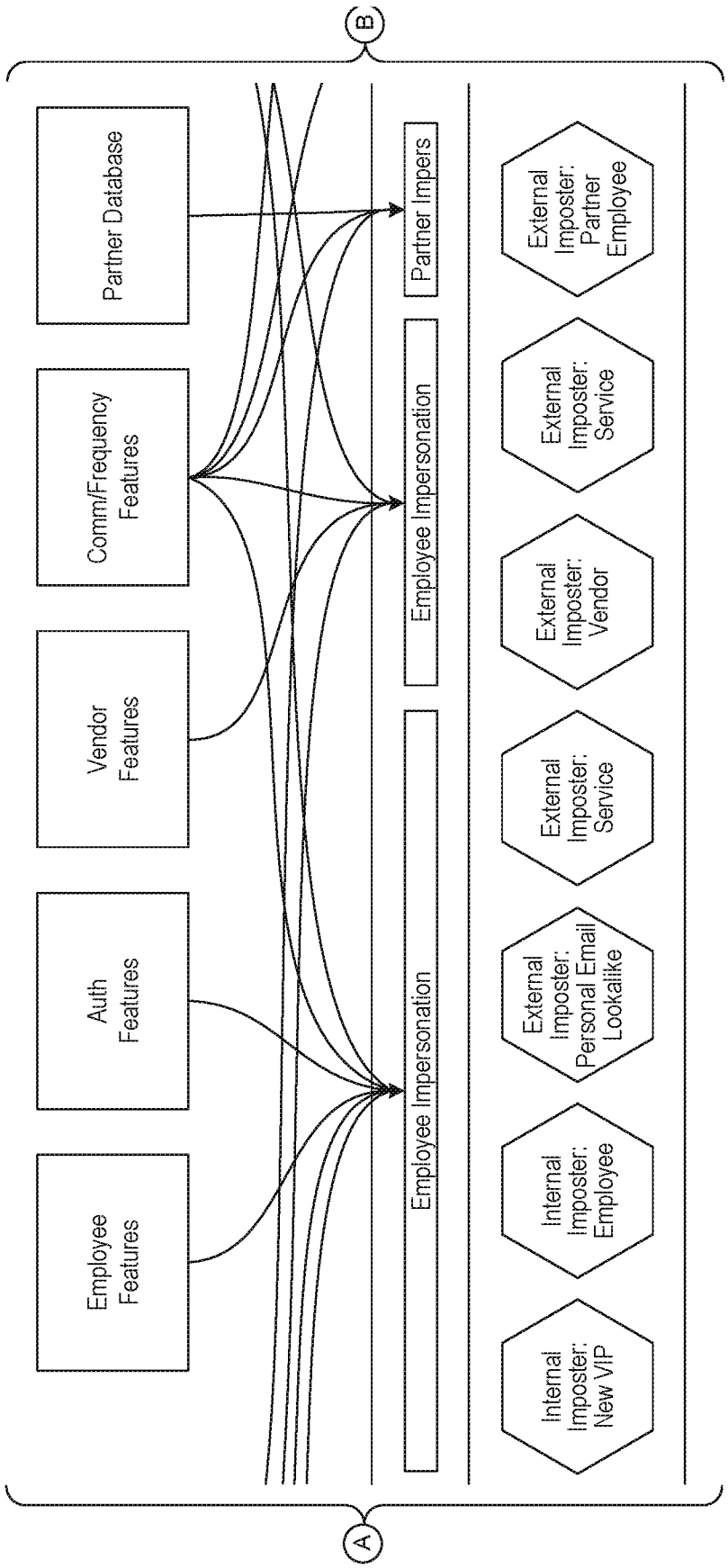
B
Partner Database
Comm/Frequency Features
Vendor Features
Auth Features
Employee Features
Partner Impers
Employee Impersonation
Employee Impersonation
External Imposter: Partner Employee
External Imposter: Service
External Imposter: Vendor
External Imposter: Service
External Imposter: Personal Email Lookalike
Internal Imposter: Employee
Internal Imposter: New VIP
A

FIG. 16, Cont.

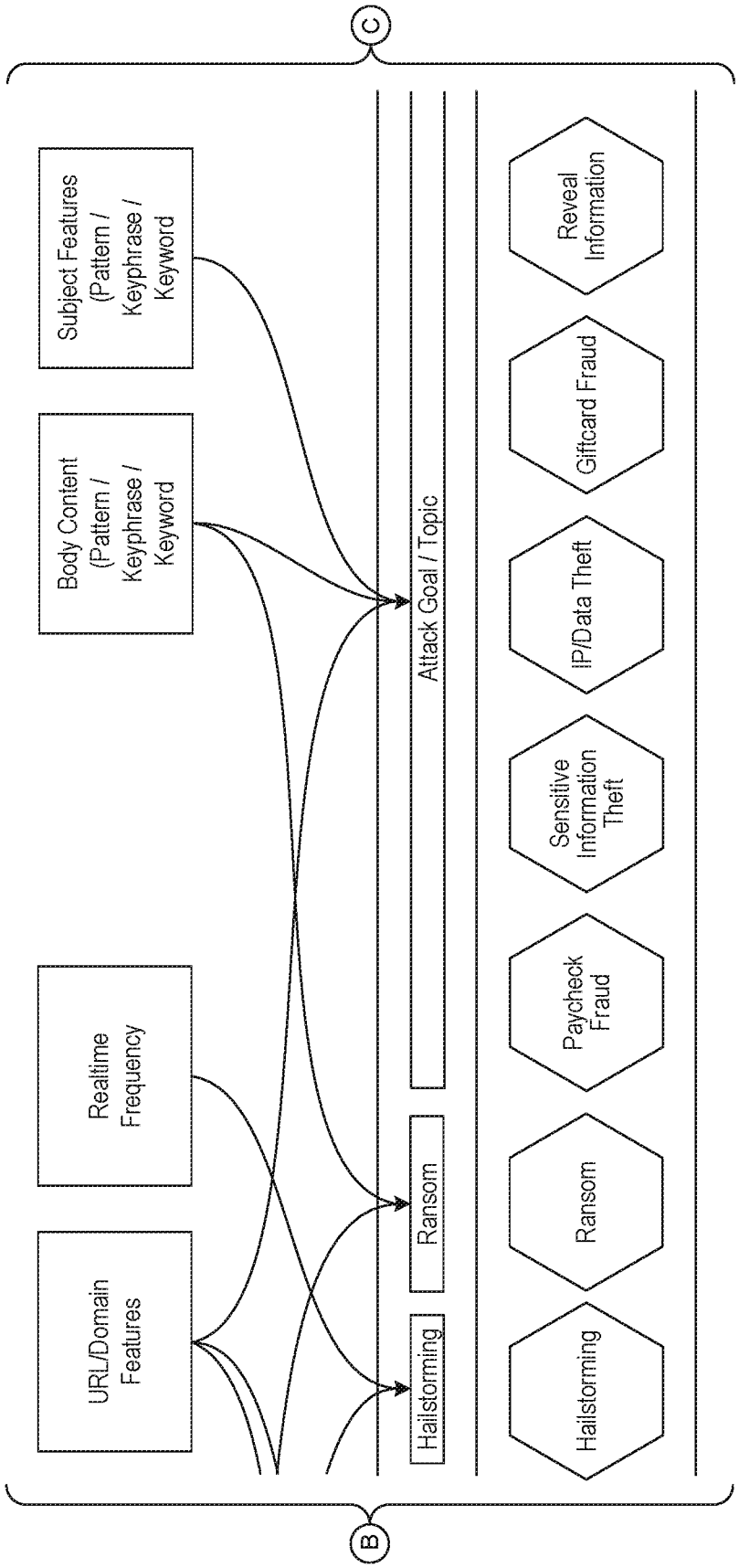
C
Subject Features (Pattern / Keyphrase / Keyword
Body Content (Pattern / Keyphrase / Keyword
Realtime Frequency
URL/Domain Features
Attack Goal / Topic
Hailstorming
Ransom
Hailstorming
Ransom
Paycheck Fraud
Sensitive Information Theft
IP/Data Theft
Giftcard Fraud
Reveal Information
B

FIG. 16, Cont. 2

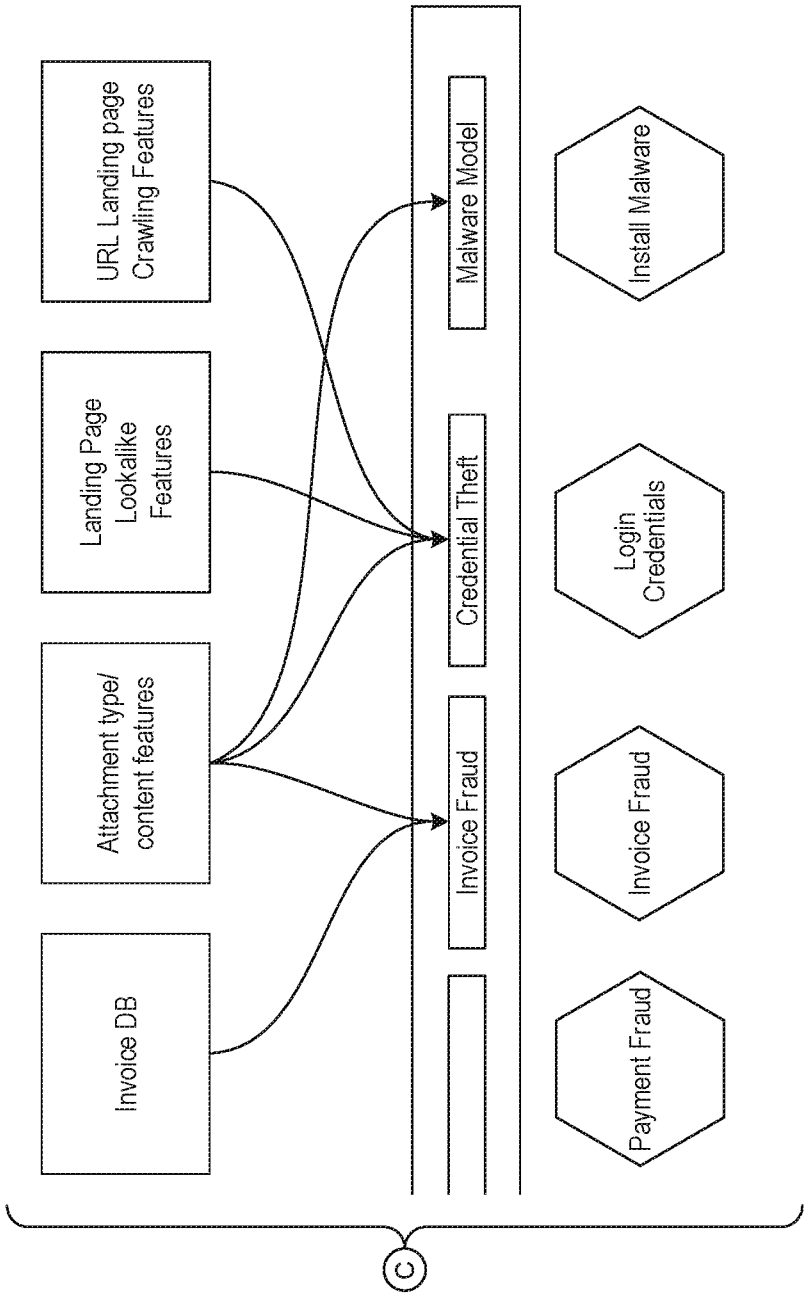
URL Landing page Crawling Features
Landing Page Lookalike Features
Attachment type/ content features
Invoice DB
Malware Model
Credential Theft
Invoice Fraud
Install Malware
Login Credentials
Invoice Fraud
Payment Fraud
C

FIG. 16, Cont. 3

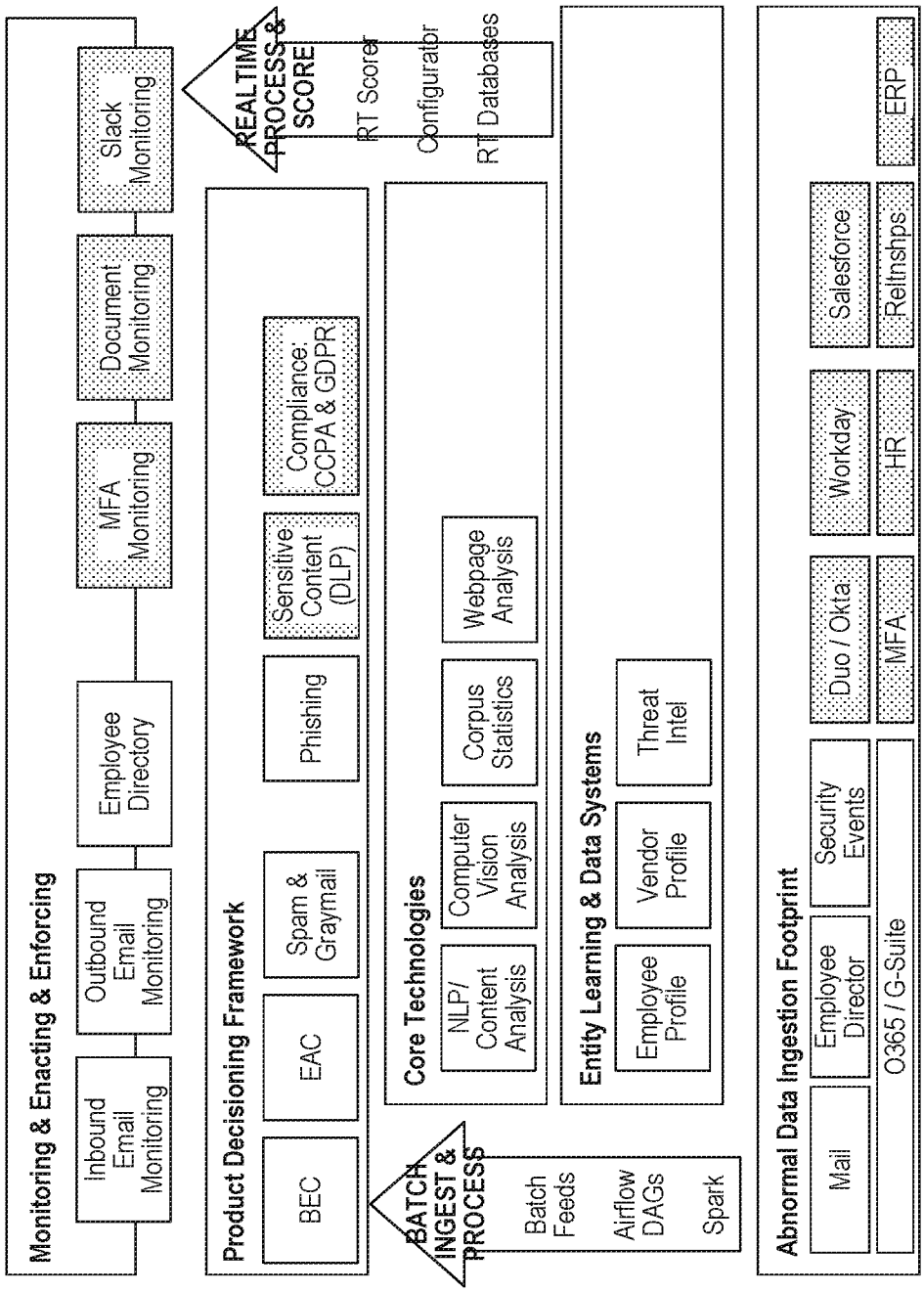
Monitoring & Enacting & Enforcing
Inbound Email Monitoring
Outbound Email Monitoring
Employee Directory
MFA Monitoring
Document Monitoring
Slack Monitoring
REALTIME PROCESS & SCORE
RT Scoref
Configuratbr
RT Databases
Product Decisioning Framework
BEC
EAC
Spam & Graymail
Phishing
Sensitive Content (DLP)
Compliance: CCPA & GDPR
BATCH INGEST & PROCESS
Batch Feeds
Airflow DAGs
Spark
Core Technologies
NLP/ Content Analysis
Computer Vision Analysis
Corpus Statistics
Webpage Analysis
Entity Learning & Data Systems
Employee Profile
Vendor Profile
Threat Intel
Abnormal Data Ingestion Footprint
Mail
Employee Director
Security Events
O365 / G-Suite
Duo / Okta
MFA
Workday
HR
Salesforce
Reltnshps
ERP

FIG. 17

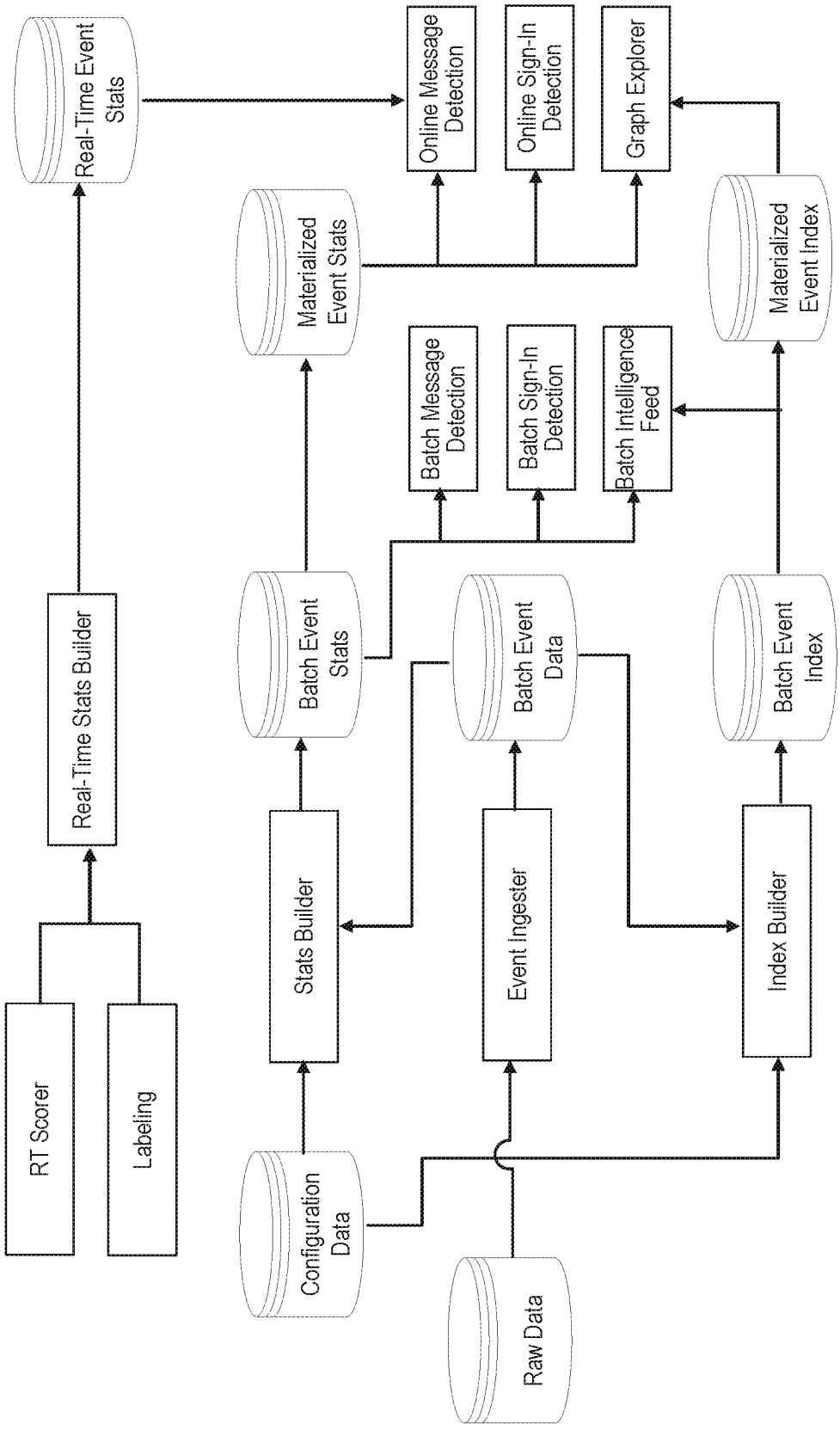
Real-Time Event Stats
Online Message Detection
Online Sign-In Detection
Graph Explorer
Materialized Event Stats
Materialized Event Index
Batch Message Detection
Batch Sign-In Detection
Batch Intelligence Feed
Real-Time Stats Builder
Batch Event Stats
Batch Event Data
Batch Event Index
RT Scorer
Labeling
Stats Builder
Event Ingester
Index Builder
Configuration Data
Raw Data

FIG. 18A

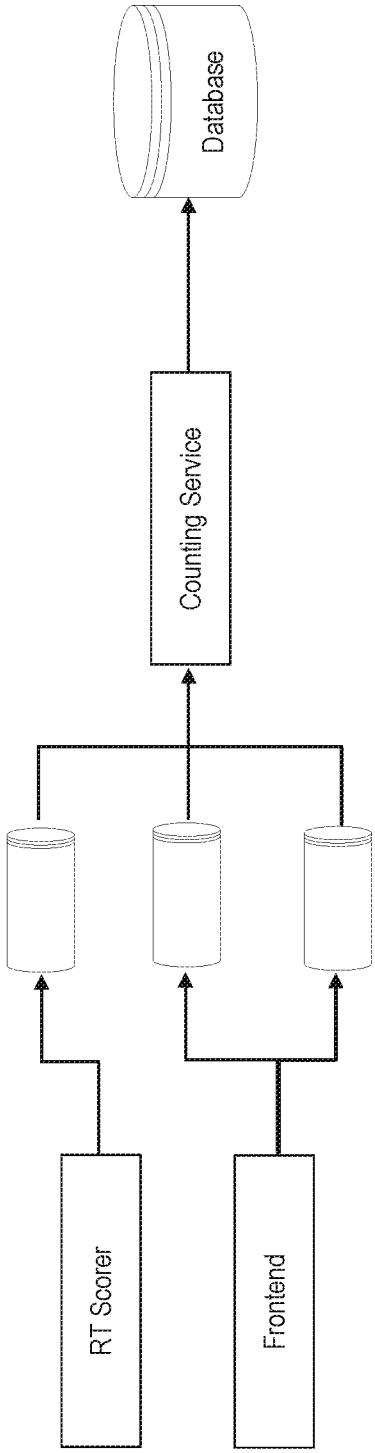

FIG. 18B

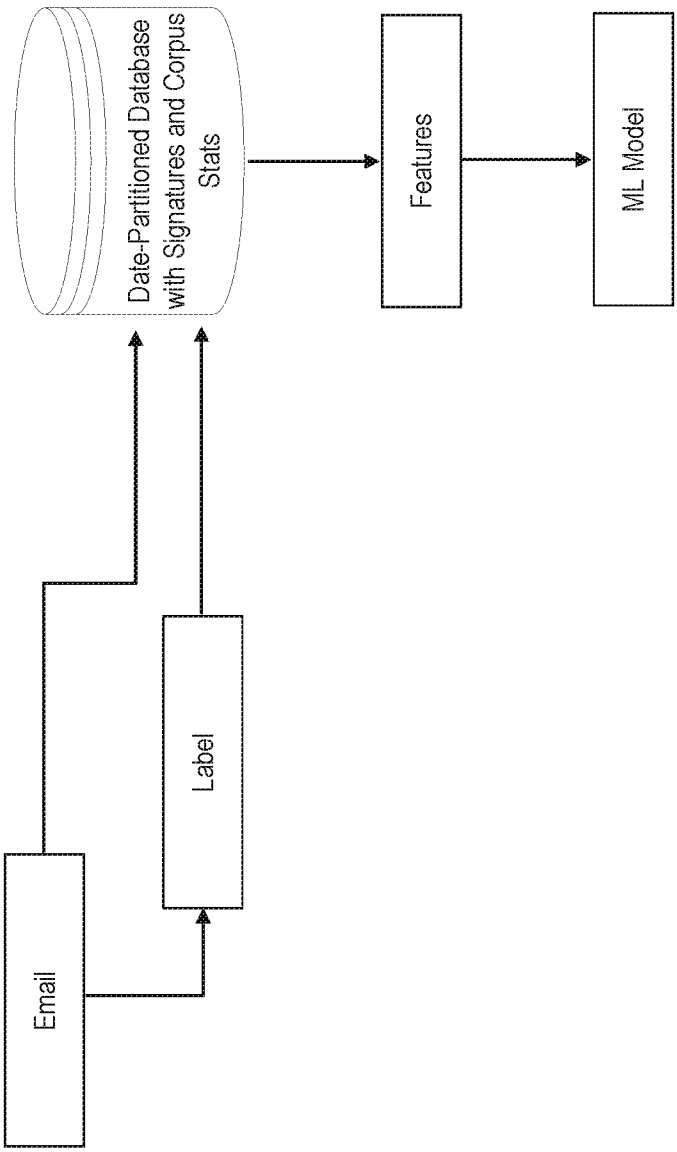
Email
Label
Date-Partitioned Database with Signatures and Corpus Stats
Features
ML Model

FIG. 19A

Entity Type	Entity	Date Partition	Safe Count	Attack Count
Link Domain	dropbox.com	Mar 12th 2019	3894329	329
Link Domain	sharepoint.com	Mar 12th 2019	93234323	8397
Link Domain	instagram.com	Mar 12th 2019	9834792	0
From Domain x Simple Auth	gmail.com x Pass	Mar 12th 2019	38294729	9837
From Domain x Simple Auth	instagram.com x Fail	Mar 12th 2019	0	390
From Email	sakura.ne.jp	Mar 12th 2019	0	5873
Subject	Update your password NOW	Mar 12th 2019	73	1329

FIG. 19B

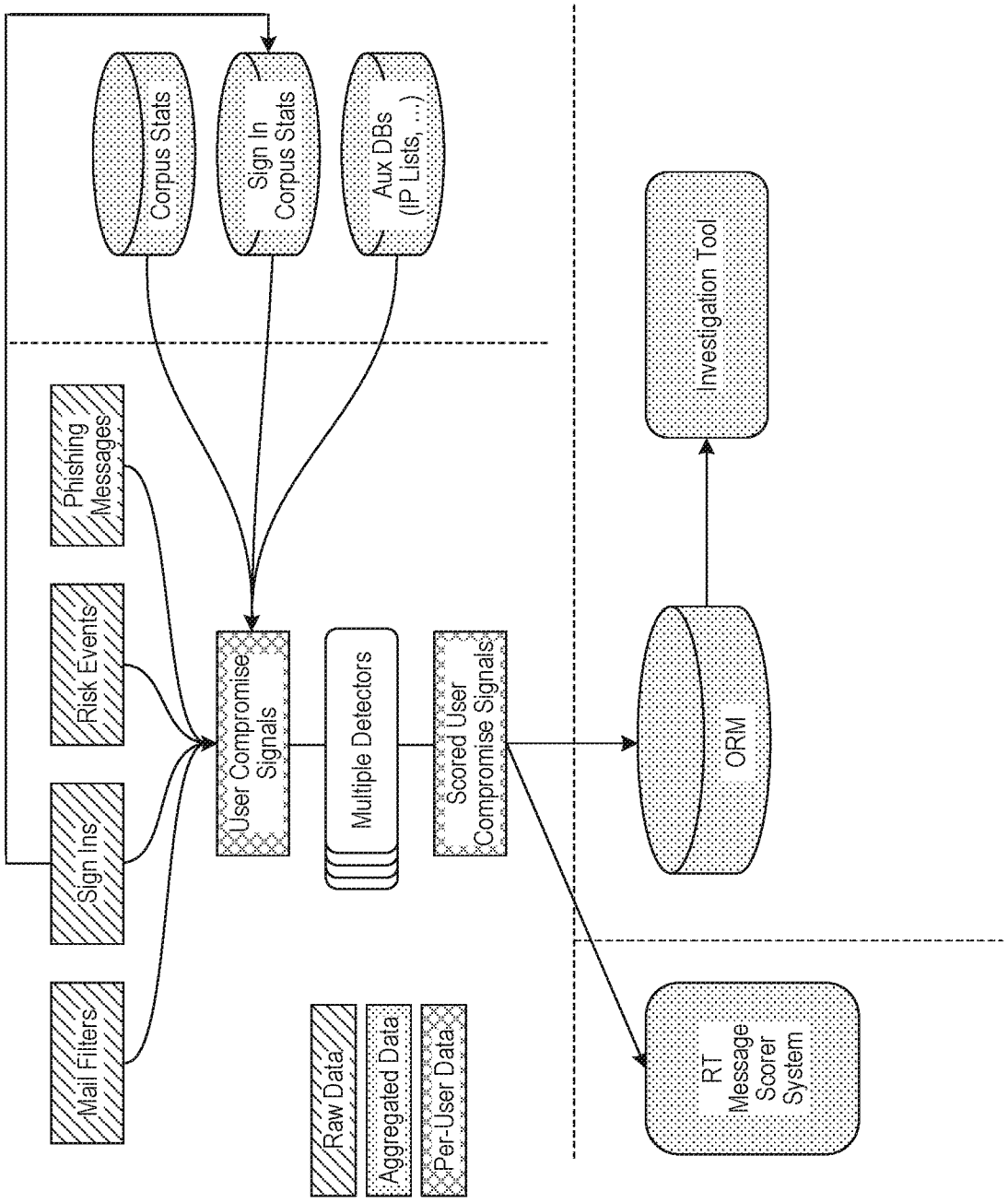
Corpus Stats
Sign In Corpus Stats
Aux DBs (IP Lists, ...)
Phishing Messages
Risk Events
Sign Ins
Mail Filters
User Compromise Signals
Multiple Detectors
Scored User Compromise Signals
Investigation Tool
ORM
RT Message Scorer System
Raw Data
Aggregated Data
Per-User Data

FIG. 20

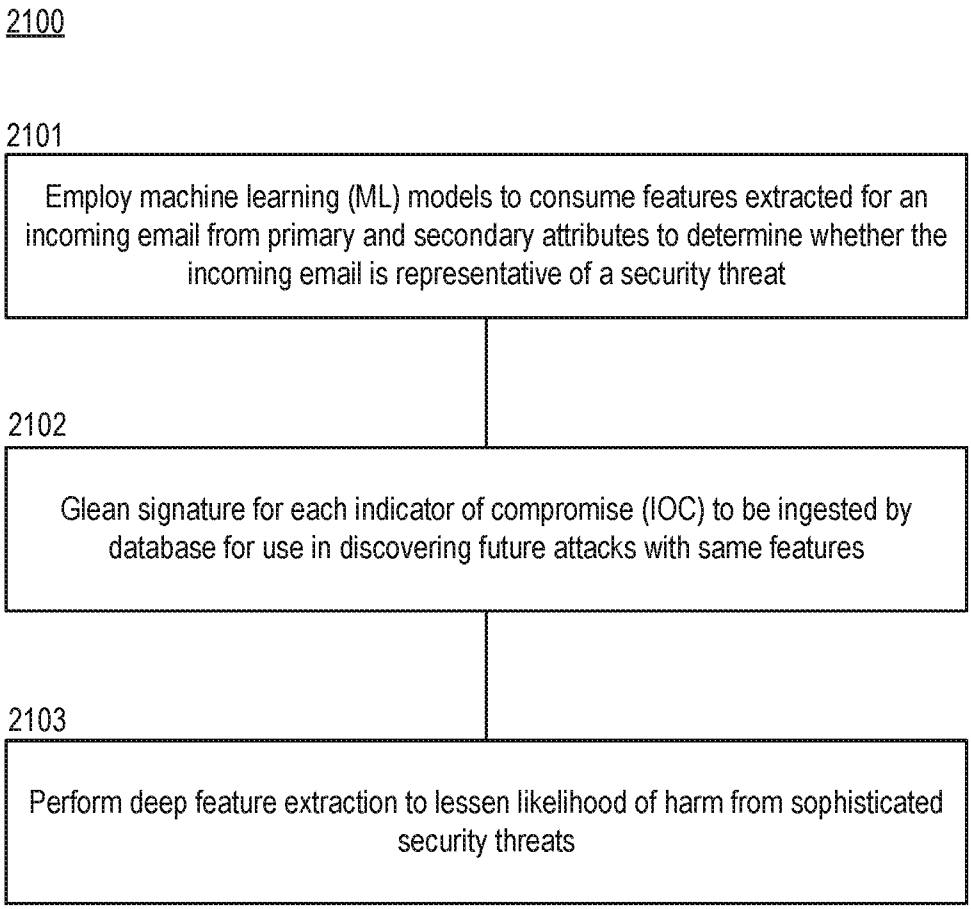
2100
2101
Employ machine learning (ML) models to consume features extracted for an incoming email from primary and secondary attributes to determine whether the incoming email is representative of a security threat
2102
Glean signature for each indicator of compromise (IOC) to be ingested by database for use in discovering future attacks with same features
2103
Perform deep feature extraction to lessen likelihood of harm from sophisticated security threats

FIG. 21

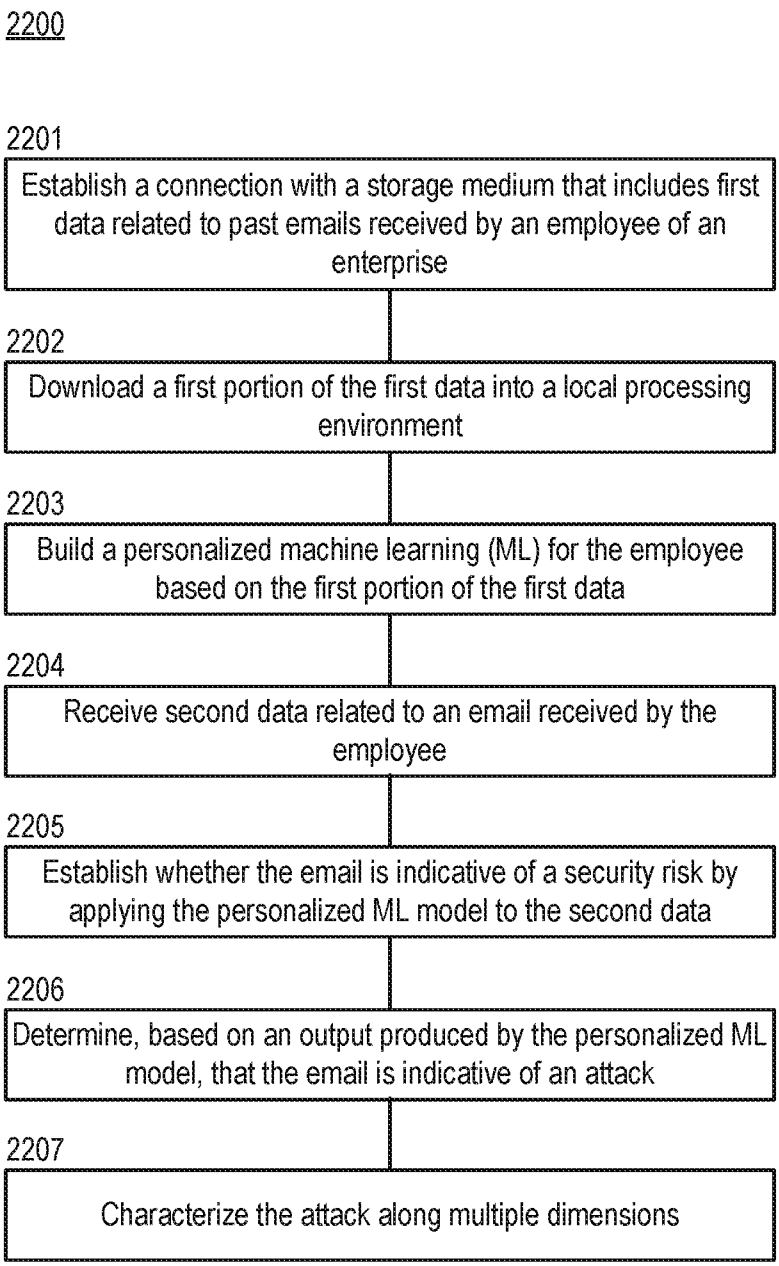
2200
2201
Establish a connection with a storage medium that includes first data related to past emails received by an employee of an enterprise
2202
Download a first portion of the first data into a local processing environment
2203
Build a personalized machine learning (ML) for the employee based on the first portion of the first data
2204
Receive second data related to an email received by the employee
2205
Establish whether the email is indicative of a security risk by applying the personalized ML model to the second data
2206
Determine, based on an output produced by the personalized ML model, that the email is indicative of an attack
2207
Characterize the attack along multiple dimensions

FIG. 22

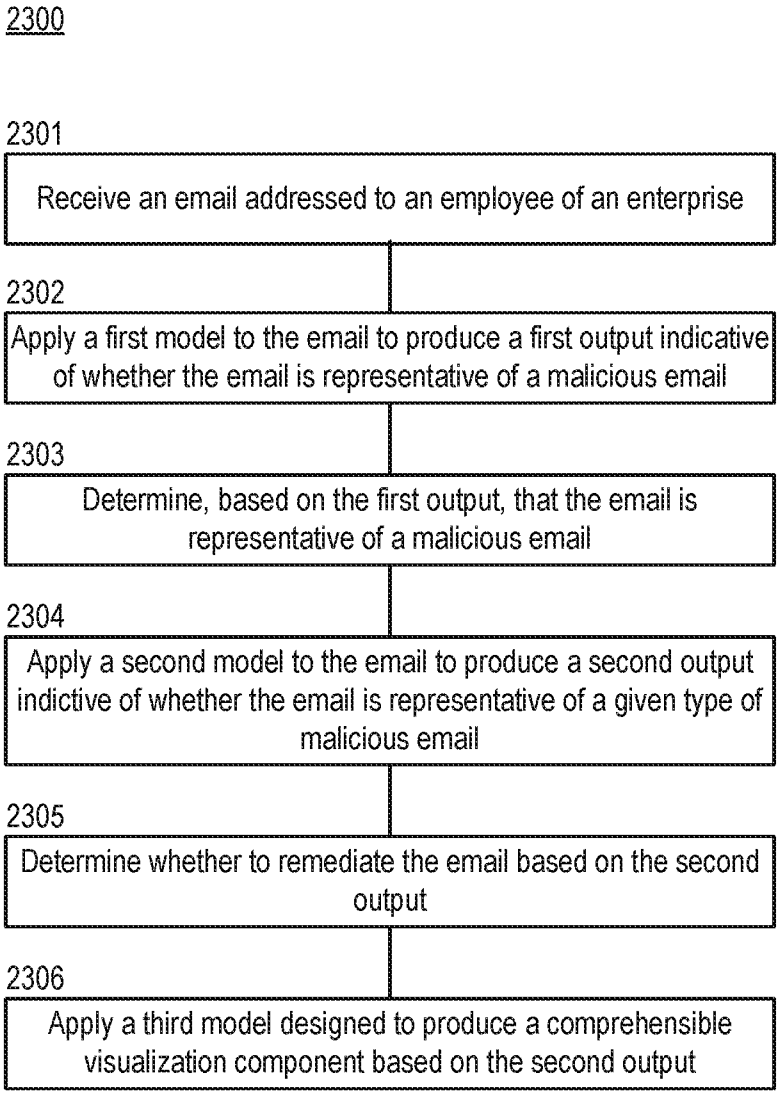
2300
2301
Receive an email addressed to an employee of an enterprise
2302
Apply a first model to the email to produce a first output indicative of whether the email is representative of a malicious email
2303
Determine, based on the first output, that the email is representative of a malicious email
2304
Apply a second model to the email to produce a second output indictive of whether the email is representative of a given type of malicious email
2305
Determine whether to remediate the email based on the second output
2306
Apply a third model designed to produce a comprehensible visualization component based on the second output

FIG. 23

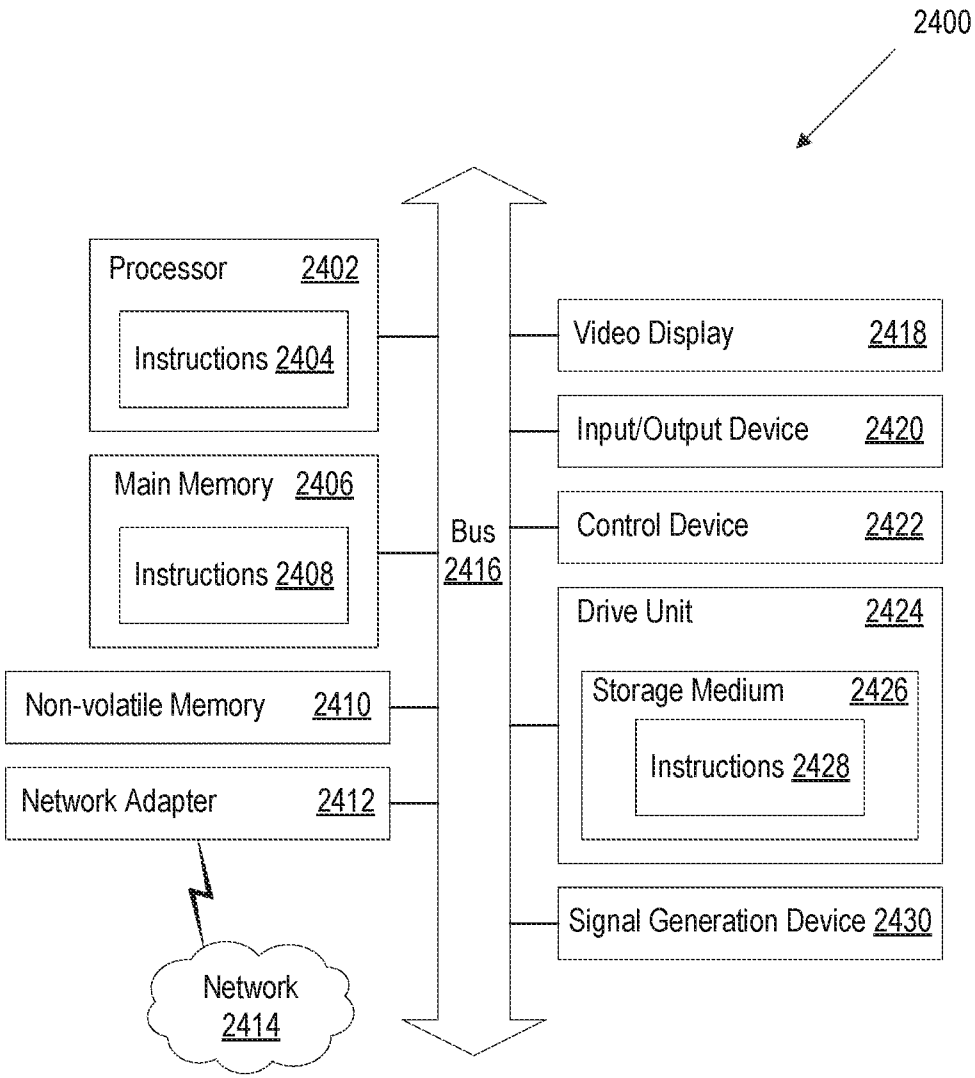
2400
Processor 2402
Instructions 2404
Main Memory 2406
Instructions 2408
Non-volatile Memory 2410
Network Adapter 2412
Network 2414
Bus 2416
Video Display 2418
Input/Output Device 2420
Control Device 2422
Drive Unit 2424
Storage Medium 2426
Instructions 2428
Signal Generation Device 2430

FIG. 24