(12) STANDARD PATENT (11) Application No. **AU 2022370400 B2**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
USER ENTITY NORMALIZATION AND ASSOCIATION

(51) International Patent Classification(s)
G06F 21/55 (2013.01) ***H04L 9/40*** (2022.01)

(21) Application No: **2022370400** (22) Date of Filing: **2022.10.06**

(87) WIPO No: **WO23/067425**

(30) Priority Data

(31) Number	(32) Date	(33) Country
17/505,673	**2021.10.20**	**US**

(43) Publication Date: **2023.04.27**
(44) Accepted Journal Date: **2024.09.12**

(71) Applicant(s)
PALO ALTO NETWORKS (ISRAEL ANALYTICS) LTD.

(72) Inventor(s)
RIMER, Netanel;MEYER, Aviad;NEUMAN, Yaron;ALLON, Jonathan

(74) Agent / Attorney
FPA Patent Attorneys Pty Ltd, Level 19, South Tower 80 Collins Street, Melbourne, VIC, 3000, AU

(56) Related Art
US 2007/0073519 A1

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 April 2023 (27.04.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/067425 A1

(51) International Patent Classification:
G06F 21/55 (2013.01) *H04L 9/40* (2022.01)

(21) International Application Number:
PCT/IB2022/059544

(22) International Filing Date:
06 October 2022 (06.10.2022)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
17/505,673 20 October 2021 (20.10.2021) US

(71) Applicant: **PALO ALTO NETWORKS (ISRAEL ANALYTICS) LTD.** [IL/IL]; 94A Yigal Alon Street, Alon 1 Tower, 6789155 Tel Aviv (IL).

(72) Inventors: **RIMER, Netanel**; 8 Pinsker Street, 7630810 Rehovot (IL). **MEYER, Aviad**; 17a Eshkol Street, 4534320 Hod Hasharon (IL). **NEUMAN, Yaron**; 65 Hagefen Street, 4282300 Zoran (IL). **ALLON, Jonathan**; 26 Sweden Street, 3491279 Haifa (IL).

(74) Agent: **KLIGLER & ASSOCIATES PATENT ATTORNEYS LTD.**; P.O. Box 20612, 6120601 Tel Aviv (IL).

(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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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, ME, 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))*

(54) Title: USER ENTITY NORMALIZATION AND ASSOCIATION

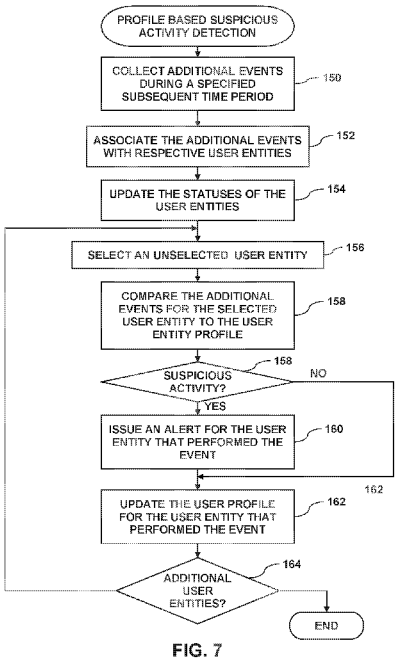

FIG. 7

(57) Abstract: Methods, apparatuses and computer program products implement embodiments of the present invention that include protecting a computer system by identifying multiple user identifiers (68) associated with a single user entity. A first event carried out using a first one of the user identifiers is detected. Upon detecting a second event carried out using a second one of the user identifiers that is different from the first one of the user identifiers, an alert can be issued in response to a combination of the first and the second events.

WO 2023/067425 A1

USER ENTITY NORMALIZATION AND ASSOCIATION

FIELD OF THE INVENTION

[0001] The present invention relates generally to computer security and networks, and particularly to associating user identifiers in event logs with a user entity and generating a user entity profile based on events in the logs.

BACKGROUND OF THE INVENTION

[0002] In many computers and network systems, multiple layers of security apparatus and software are deployed in order to detect and repel the ever-growing range of security threats. At the most basic level, computers use anti-virus software to prevent malicious software from running on the computer. At the network level, intrusion detection and prevention systems analyze and control network traffic to detect and prevent malware from spreading through the network.

[0003] The description above is presented as a general overview of related art in this field and should not be construed as an admission that any of the information it contains constitutes prior art against the present patent application.

SUMMARY OF THE INVENTION

[0003A] According to a first aspect of the invention there is provided a method for protecting a computer system, comprising: identifying, by a processor, multiple user identifiers associated with a single user entity; detecting status of the user entity, based on an information record of a first one of the user identifiers, wherein the status has a timespan; detecting an event carried out during the timespan using a second one of the user identifiers that is not in accordance with the status of the user entity based on the information record of the first one of the user identifiers; and issuing an alert in response to detecting the event carried out during timespan.

[0003B] According to a second aspect of the invention there is provided an apparatus for protecting a computer network, comprising: a network interface card (NIC); and at least one processor configured: to identify multiple user identifiers associated with a single user entity, to detect status of the user entity, based on an information record of a first one of the user identifiers, wherein the status has a timespan, to detect an event carried out during the timespan using a second one of the user identifiers that is not in accordance with the status of the user entity based on the information record of the first one of the user identifiers, and to issue an alert in response to detecting the event carried out during the timespan.

2022370400 22 Aug 2024

[0003C] According to a third aspect of the invention there is provided a computer software product for protecting a computing system, the product comprising a non-transitory computer-readable medium, in which program instructions are stored, which instructions, when read by a computer, cause the computer: to identify multiple user identifiers associated with a single user entity; to detect a status of the user entity, based on an information record of a first one of the user identifiers, wherein the status has a timespan; to detect an event carried out during the timespan using a second one of the user identifiers that is not in accordance with the status of the user entity based on the information record of the first one of the user identifiers; and to issue an alert in response to detecting the event carried out during the timespan.

[0003D] By way of clarification and for avoidance of doubt, as used herein and except where the context requires otherwise, the term "comprise" and variations of the term, such as "comprising", "comprises" and "comprised", are not intended to exclude further additions, components, integers or steps.

[0004] There is provided, in accordance with an embodiment of the present invention, a method for protecting a computer system, including identifying, by a processor, multiple user identifiers associated with a single user entity, detecting a first event carried out using a first one of the user identifiers, detecting a second event carried out using a second one of the user identifiers that is different from the first one of the user identifiers, and in response to a combination of the first and the second events, issuing an alert.

[0005] In some embodiments, identifying the multiple user identifiers associated with the single user entity includes collecting a set of events including the first and the second events, extracting respective user identifiers from the events in the set, mapping the extracted user identifiers to respective accounts, and associating the accounts with respective user entities, wherein the single user entity includes one of the multiple user entities.

[0006] In a first embodiment, mapping a given extracted user identifier to a given account includes normalizing the given user entity to a specific format, wherein the given account includes the normalized user entity.

[0007] In a second embodiment, the single user entity is associated with one or more accounts.

[0008] In a third embodiment, multiple user identifiers map to a given account for the single user entity.

2022370400 22 Aug 2024

[0009] In additional embodiments, detecting the first event includes detecting the first event on a first networked entity, and wherein detecting the second event includes detecting the second event on a second networked entity different from the first networked entity.

[0010] In further embodiments, detecting the first even includes detecting multiple first events during a first time period, and the method further includes generating a profile in response to the multiple first events, wherein detecting a second event includes detecting one or more second events in a second time period subsequent to the first time period, and wherein the combination of the first and the second events includes detecting that the one or more second events are not in accordance with the profile.

[0011] In supplemental embodiments, the first event includes a time-based status of the single user entity, and wherein the second event is not in accordance with the time-based status.

[0012] There is also provided, in accordance with an embodiment of the present invention, an apparatus for protecting a computer network, including a network interface card (NIC), and at least one processor configured to identify multiple user identifiers associated with a single user entity, to detect a first event carried out using a first one of the user identifiers, to detect a second event carried out using a second one of the user identifiers that is different from the first one of the user identifiers, and in response to a combination of the first and the second events, to issue an alert.

[0013] There is additionally provided, in accordance with an embodiment of the present invention, a computer software product for protecting a computing system, the product including a non-transitory computer-readable medium, in which program instructions are stored, which instructions, when read by a computer, cause the computer to identify multiple user identifiers associated with a single user entity, to detect a first event carried out using a first one of the user identifiers, to detect a second event carried out using a second one of the user identifiers that is different from the first one of the user identifiers, and in response to a combination of the first and the second events, to issue an alert.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The disclosure is herein described, by way of example only, with reference to the accompanying drawings, wherein:

[0015] Figure 1 is a block diagram that schematically shows a computing facility comprising a security server that is configured to generate activity profiles for user entities based on events retrieved from network event logs, in accordance with an embodiment of the present invention;

[0016] Figure 2 is a block diagram showing an example of a given event log, in accordance with an embodiment of the present invention;

[0017] Figure 3 is a block diagram showing an example of an aggregated event log stored on the security server, in accordance with an embodiment of the present invention;

[0018] Figure 4 is a block diagram showing an example of a database record that can be stored by a domain database server, in accordance with an embodiment of the present invention;

[0019] Figure 5 is a block diagram showing an example of user entity information stored on the security server, in accordance with an embodiment of the present invention;

[0020] Figure 6 is a flow diagram that schematically illustrates a method of generating the activity profiles, in accordance with an embodiment of the present invention; and

[0021] Figure 7 is a flow diagram that schematically illustrates a method of using the generated activity profiles to detect suspicious activity, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF EMBODIMENTS

[0022] Networked entities that communicate over computer network typically store logs that record events on the networked entities. While these logs can include identifiers for the events, user entities (e.g., employees of an organization) may use multiple accounts (e.g., email accounts) when accessing data on the network, and each account may use multiple identifiers when accessing data on the network. Therefore, it can be difficult to detect suspicious/malicious activity performed by a given user entity using different accounts and different user identifiers on the network.

[0023] Embodiments of the present invention provide methods and systems for protecting a computer system by identifying multiple user identifiers associated with a single user entity. Upon detecting a first event carried out using a first one of the user identifiers and detecting a second event carried out using a second one of the user identifiers that is different from the first one of the user identifiers, an alert can be issued in response to a combination of the first and the second

events. In some embodiment the first event is collected from a first log on a first networked entity and the second event is collected from a second log on a second networked entity different than the first networked entity.

[0024] In one embodiment, multiple events can be collected from multiple event logs on networked entities coupled to a computer network. Identifiers can be extracted from the events, the identifiers can be normalized so as to map the events to accounts, and a subset of the accounts can be associated with the single user entity. A user entity profile can then be generated based on the events associated with the single user entity. Using this embodiment, systems implementing embodiments of the present invention can detect and flag suspicious activity if any subsequent events associated with the single user entity are determined not to be in accordance with the user entity profile.

SYSTEM DESCRIPTION

[0025] Figure 1 is a block diagram that schematically shows an example of a computing facility 20 comprising a security server 22 that is configured to generate user entity activity profiles 24 based on activity recorded by a plurality of networked entities in respective event logs 26, in accordance with an embodiment of the present invention. In the configuration shown in Figure 1, security server 22 is configured to communicate with a plurality of computing devices 28 (also known as hosts or host computers), an account database server 29 and a human resources (HR) server 30 over a data network such as a local area network (LAN) 32.

[0026] Account database server 29 may comprises a domain database management system (DBMS) application 31 and a domain database 37. Account database 33 comprises a set of account database records 35 that are described in the description referencing Figure 4 hereinbelow.

[0027] Computing facility 20 may also comprise an Internet gateway 34, which couples computing facility 20 to a public network 36 such as the Internet. To protect computing devices 28, computing facility 20 may also comprise a firewall 38 that is coupled to LAN 32 and controls, based on predetermined security rules, data traffic between computing devices 28 and a data cloud 40 comprising one or more cloud servers 42.

[0028] As described supra, security server 22 can be configured to generate user entity profiles 24 based on activity recorded by a plurality of networked entities in respective event logs 26. While the configuration in Figure 1 shows the networked entities comprise computing devices 28, firewall 38 and cloud servers 42, any other type of networked entities that communicate over a network are considered to be within the spirit and scope of the present invention.

[0029] In the configuration shown in Figure 1, event logs 26 can be differentiated by appending a letter to the identifying numeral, so that the web pages:

- Operating system (OS) logs 26A store information on events generated by operating systems (such as Windows™ produced by Microsoft Corporation, and Linux™) and applications executing on computing devices 28.
- Endpoint detection and response (EDR) logs 26B store information on events detected by endpoint agents 44 (e.g., XDR™ produced by Palo Alto Networks, Inc., of 3000 Tannery Way, Santa Clara, CA 95054 USA) executing on computing devices 28.
- Firewall log 26C stores information on transmissions between computing facility 20 (e.g., computing devices 28) and servers (e.g., cloud servers 42) coupled to Internet 36. One example of a firewall 38 is the PA-3250 Next Generation Firewall™ produced by Palo Alto Networks, Inc.
- Cloud event logs 26D store information on events generated by cloud servers 42. Examples of logs 26 include, but are not limited to application logs, resource logs, and service logs for Amazon Web Services (provided by Amazon.com, Inc., 410 Terry Avenue North Seattle, WA 98109 USA).

[0030] In embodiments described herein, security server 22 can be configured to extract user identifiers (IDs) from logs 26, normalize the user IDs and associate the normalized user IDs with user entities (i.e., individual people such as employees). In some embodiments, HR server 30 stores an HR database 46 that stores information for each user entity. In some embodiments, HR database 46 comprises a set of records 47 that have a one-to-one correspondence with user entities (i.e., employees) of an organization.

[0031] Security server 22 comprises a processor 48, a memory 50 and a network interface card (NIC) 51 that couples the security server to LAN 32. In some embodiments, processor 48 can combine logs 26 into an aggregated event log 52. Event logs 26 and 52 are described respectively in the descriptions referencing Figure 2 and 3 hereinbelow.

[0032] While in embodiments described herein, processor 48 collects events from event logs 24A-24D, and stores to aggregated event log 52, aggregating events from other types of event logs 26 into the aggregated event log is considered to be within the spirit and scope of the present invention. Examples of information that can be stored by one or more additional event logs 26 include, but are not limited to:

- Input/Output (I/O) events (also known as file events). An example of an I/O event is domain account "*Company\jdoe*" writing a file named "*local_file\malicious.exe*". Domain accounts are described hereinbelow.
- Registry events. An example of a registry event is domain account "*Company\jdoe*" modifying a registry key related to *Autorun*, with the value "*local_file\malicious.exe*".
- Process execution events. An example of a process execution event is *SYSTEM* automatically executing "*local_file\malicious.exe*" with permissions of domain account "*company\jdoe*".
- Network events. An example of a network event is domain account "*company\jdoe*", using a process named local_file\malicious.exe, performed an HTTP request to "*www.malware_command_and_control*.com".
- Single sign-on (SSO) events. SSO services (e.g., *Okta*™, *PingOne*™, *AzureAD*™) typically provide audit logs, which. One example of an event that processor 48 can collect is SSO account "*john.doe@company.com*" logging in.
- Email events. Email logs that store email events can be collected from local systems such as *Outlook*™, server (i.e., corporate) systems such as *Exchanger Server*™ and cloud-based email servers such as *Exchange Online*™. An example of an email event in a local system is an email sent/received by "*john.doe@gmail.com*". An example of an email event in a server system is an email sent/received by "*johndoe@company.com*". An example of an email event in a cloud-based system is an email sent/received by "*john_doe@cloud_email_provider.com*".

[0033] In the configuration shown in Figure 1, memory 50 also stores a plurality of user entity records 54 that store profiles 24. In some embodiments, each given user entity record 54 can retrieve information for a given user entity from HR database 46 and store the retrieved information to the given user entity record.

[0034] In some embodiments, tasks described herein such as extracting user-IDs from event logs 26, normalizing the user IDs, associate the normalized user IDs with user entities, aggregating logs 26 into aggregated event log 52, and generating user entity profiles 24 may be split among multiple computers systems 22, 28 and 30 within computing facility 20 or external to the computing facility (e.g., cloud servers 42). In additional embodiments, the functionality of some

or all of computing devices 28, security server 22, account database server 29 and HR server 30 may be deployed in computing facility 20 and/or Internet 36 as physical computing devices, virtual machines or containers.

[0035] In some embodiments, client computers 28 have respective host names 56 that can be used to identify each of the client computers.

[0036] Processor 48 comprises a general-purpose central processing units (CPU) or special-purpose embedded processors, which are programmed in software or firmware to carry out the functions described herein. This software may be downloaded to security server 22 in electronic form, over a network, for example. Additionally or alternatively, the software may be stored on tangible, non-transitory computer-readable media, such as optical, magnetic, or electronic memory media. Further additionally or alternatively, at least some of the functions of processor 48 may be carried out by hard-wired or programmable digital logic circuits.

[0037] Examples of memory 50 include dynamic random-access memories, non-volatile random-access memories, hard disk drives and solid-state disk drives.

[0038] Figure 2 is a block diagram shown an example of data components stored in event logs 26, in accordance with an embodiment of the present invention. While event logs 26A-26D may store information in different respective layouts (i.e., formats and schemas), for purposes of simplicity the event logs herein comprise a single layout.

[0039] In the example shown in Figure 2, each event log 26 comprise a set of event log entries 60, each of the event log entries comprising a date 62, a time 64 and an event message 66 that stores a description of an event. For a given event in a given event log entry 60, date 62 comprises the date of the given event, time 64 comprises the time of the given event and event message 66 describes an event and lists user identifiers of participants. User identifiers are described in the description referencing Figure 3 hereinbelow.

[0040] Each event message 66 (i.e., referencing a given event) can have one or more user identifiers 68 (i.e., participants in the corresponding event). In one example, if a given event message corresponds to an event comprising a user entity sending an email, then the given event message 66 comprises a single identifier (ID) 68. In another example, if a given event message corresponds to an event comprising a first account associated with a first user entity granting one or more system permissions to a second account associated with a second user entity, then the given event message may comprise two identifiers 68.

[0041] In embodiments of the present invention, there are multiple user entities 67 (i.e., individual physical users) that operate computing devices using one or more respective accounts 69. As described hereinbelow, processor 48 can map each identifier 68 to a respective account 69, and then associate each account 69 with a respective user entity 67. Accounts 69 are described in the description referencing Figure 5 hereinbelow.

[0042] In some embodiments, processor 48 can retrieve event log entries 60 from all the event logs (e.g., event logs 26A-26D), and store event information in the retrieved event log entries to aggregated event log 52. As described hereinbelow, processor 48 can use the information stored in aggregated event log 52 to map events to user entities.

[0043] Figure 3 is a block diagram shown an example of data components stored in aggregated event log 52, in accordance with an embodiment of the present invention. Aggregated event log 52 comprises a set of aggregated log entries 70. In some embodiments, processor 48 can create a new aggregated log entry 70 for each event log entry 60 in each event log 26. In other words, each aggregated log entry 70 has a corresponding event log entry 60.

[0044] Each aggregated event log entry 70 comprises an event ID 72, a source 74, a date 76, a time 78, an event message 80 and an identifier information record 82. Upon creating a new aggregated log entry 70 for a corresponding event log entry 60, processor 48 can:

- Create a unique event ID 72.
- Store, to source 74, an identifier of the device that generated the event log storing the corresponding event log entry 60. Examples of identifiers include, but are not limited to, Internet Protocol (IP) address of a given cloud server 42 or a media access control (MAC) address of a given computing device 28.
- Copy date 62 in the corresponding event log entry 60 to date 76.
- Copy time 64 in the corresponding event log entry 60 to time 78.
- Copy event message 66 in the corresponding event log entry 60 to event message 80.

[0045] In some embodiments, processor 48 can extract one or more user IDs 69 from event message 80s, normalize the user IDs and associate the normalized user IDs with user entities. In the configuration shown in Figure 3, each user ID 69 extracted from a given event message 80 has a corresponding identifier record 82 that stores information such as an extracted user identifier 84, an identifier type 86, a mapped account m and an associated user entity 90.

[0046] Upon creating the new aggregated log entry (i.e., as described supra), processor 48 can identify a number (i.e., one or more) identifiers 68 in event message 80, add the identified number of identifier information records 82 to the new aggregated log entry so that each identifier 68 has a corresponding identifier information record 82, and populate each given identifier information record as follows:

- Store the corresponding identifier 68 to extracted identifier 84.
- Classify extracted identifier 84, and store the classification to identifier type 86. Identifier classifications are described hereinbelow.
- Normalize the corresponding identifier 68 so as to map the corresponding identifier to a given account 69, and store the mapped account to mapped account 88. Normalizing identifiers 68 is described in the description referencing Figure 6 hereinbelow.
- Identify a given user entity 67 associated with mapped account 88, and store the identified user entity to associated user entity 90. Identifying associated user entities 90 is described in the description referencing Figure 6 hereinbelow.

[0047] In examples described hereinbelow, a given user entity 67 named "*John Doe*" works for a company "*Company*", has multiple mapped accounts 88, each referenced by one or more identifiers 84.

[0048] Examples of identifier types 86 include, but are not limited to:

- Domain names such as "*Company/jdoe*". Domain names can typically be found in event messages 66 in event logs 26A, 26B and 26C.
- Fully qualified domain names (FQDN) such as "*Company.com/jdoe*". FQDNs can typically be found in event messages 66 in event logs 26A, 26B and 26C.
- A username (i.e., without a domain) such as "*jdoe*". Usernames can typically be found in event messages 66 in event logs 26A, 26B and 26C.
- A Security Identifier (SID) number such as "S-*1-5-21-1602811402-2595058921-120187713-502*". SID numbers can typically be found in event messages 66 in event logs 26A and 26B.
- A Globally Unique Identifier (GUID) number such as "*8c6bfd4a-4cb2-11ea-b67e-88e9fe502c1f*". GUID numbers can typically be found in event messages 66 in event logs 26B and 26D.

- A local username such as "*host123\jdoe*", where "*host123*" comprises a given host name 56. Local usernames can typically be found in event messages 66 in event logs 26A, 26B and 26C.
- A corporate username such as "*john.doe@company.com*". These usernames can typically be found in event messages 66 in event logs 26 such as SSO logs (not shown), email logs (not shown), and event logs 26C and 26D.
- A personal username such as "*john.doe@gmail.com*". These usernames can typically be found in event messages 66 in event logs 26 such as SSO logs (not shown), email logs (not shown), and event logs 26C and 26D.

[0049] Figure 4 is a block diagram showing an example configuration of a given database record 35, in accordance with an embodiment of the present invention. Each database record 35 can store information such as an event identifier 92 and a corresponding account identifier 94 that references a given account 69. Using this configuration, account database records 33 can store known relationships between identifiers 68 and accounts 67.

[0050] In some embodiments, account database 33 may comprise Directory Sync Service™ (DSS™), produced by Palo Alto Networks, Inc., and endpoint agents 44 may comprise XDR™, the XDR™ endpoint agent may interact with DSS™ to retrieve mappings between identifiers 68 and accounts 67.

[0051] For example, relationships between identifiers 68 and accounts 67 can be maintained by a directory services application (not shown) such as is Active Directory™ (produced by Microsoft Corporation, Redmond, Washington, USA) that performs operations such as authenticating and authorizing all users and computers in a Windows™ domain type network, assigning and enforcing security policies for all computers, and installing or updating software. In this example, account DBMS 31 can query Active Directory™ to retrieve mappings between identifiers 68 and accounts 67 that comprise domain accounts.

[0052] Figure 5 is a block diagram showing an example of information stored in user entity records 54, in accordance with an embodiment of the present invention. In the configuration shown in Figure 5, each user identity record 54 stores information such as a user entity ID 100, user entity profile 24, a set of status information records 104, a set of account information records 106 and a set of identifier-account mapping records 108.

[0053] User entity ID 100 comprise a unique identifier for a given user entity 67. In some embodiments, processor can create a set of user entity records 54 that have a one-to one

correspondence with account database records 47, and store a unique identifier to each user entity id 100 in the set. Therefore, each given user entity (i.e., employee) 67 has a corresponding user entity record 54. User entity IDs 100 may also be referred to herein as user entities 100.

[0054] User entity profile 24 comprises a user profile indicating expected activity of the corresponding user entity. As described in the description referencing Figure 7 herein below, processor 48 can use user profile 24 to detect any anomalies in actions performed by the corresponding user entity in computing facility 20.

[0055] Each status information records comprises a start date 110, and end date 112 and a status 114. Each given status 114 spans a time period starting with start date 110 and ending with end date 112. In some embodiments, start date 110 and end date 112 may also include time (e.g., 13:30 on 12/11/22).

[0056] Examples of statuses 114 include, but are not limited to:

- Employment period. Processor 48 can flag activity (e.g., emails, file access) by the corresponding user entity as suspicious if the user entity is no longer employed by the organization.
- Vacation period. Processor 48 can flag activity (e.g., emails, file access) by the corresponding user entity as suspicious if the user entity is on vacation.
- Location. Organizations may have multiple locations, and HR database can keep track of the location where each user entity works at a given time. In some embodiments, processor 48 can use this information to detect activity by a given user entity working from an anomalous location.
- Device. User entities 100 may use different computing devices 28 (e.g., desktop/laptop computers and mobile devices). Processor 48 can use this information to track which of the user entities are using which computing devices 28 at any given time (i.e., past or present)
- Department. At any given time, each user entity 100 can be assigned to a specific department (e.g., finance, marketing), thereby indicating systems (e.g., payroll, ad tracking) that are typically accessed by employees in each department.
- Title. An organization title of a given user entity 100 (e.g., manager, supervisor) can indicate privileges and typical system behavior for the given user entity.

[0057] Each user entity ID 100 typically uses one or more email accounts. In the configuration shown in Figure 5, each given user entity 100 comprises a corresponding user entity record 54 that

stores a corresponding account information record 106 for each of the email accounts used by the given user entity.

[0058] Each account information record 106 can store information such as a unique account ID 116, an account name 118 (i.e., an email address such as "*john.doe@company.com*" and john.doe@gmail.com) and account type 120. In embodiments herein, account ID 116 may also be referred to as account 116.

[0059] Examples of account types 120 include, but are not limited to:

- Domain accounts such as "*Company/jdoe*". A domain account comprises an account that can be used across Acvive Directory™ (produced by Microsoft Corporation) domain in an organization. Domain accounts are typically associated with the following identifier types 86: domain name s, FQDNs, usernames, SID numbers, GUID numbers and corporate identifiers.
- Local accounts comprising accounts such as "*host123/jdoe*" (i.e., where "host123" comprises a given host name 56) that are bound to specific respective networked entities. Local accounts are typically associated with the following identifier types 86: usernames, SID numbers, GUID numbers and local users.
- Cloud accounts such as "*john.doe@company.com*". A cloud account can be used across cloud infrastructure, like Google Cloud Platform™ (provided by Alphabet Inc., Mountain View, California) or Azure ™ (provided by Microsoft Corporation). Cloud accounts are typically associated with the following identifier types 86: GUID numbers, corporate identifiers and personal identifiers.
- Personal accounts comprising accounts such as "*john.doe@gmail.com*" that can be used both inside and outside an organization. Personal accounts are typically associated with the personal identifiers.

[0060] In embodiments of the present invention, processor 48 extracts identifiers 84 from event log entries 60 and normalizes the extracted identifiers so as to identify respective mapped accounts 88. For a given user entity 100 in the corresponding user entity record 54, processor 48 can store, in identifier-account mapping records 108, current mappings between the extracted identifiers and the associated accounts (i.e., both for the given user entity). Each identifier-account mapping record 108 in a given user entity record 54 (i.e., for a corresponding user entity 100) can store information such as:

- A user identifier 122 comprising a given identifier 84 used by the corresponding user entity.
- An identifier type 124. As described supra, identifier types 124 comprise domain names, FQDNs, usernames, SID numbers, GUID numbers, local usernames, corporate identifiers and personal identifiers.
- An associated account ID 126 that stores a given account ID 116 that processor 48 associates with identifier 122.

USER ENTITY IDENTIFICATION

[0061] Figure 6 is a flow diagram that schematically illustrates a method of associating activity in event logs 26 with user entities 100 and generating profiles 24 based on activity of the user entities in computing facility 20, in accordance with an embodiment of the present invention.

[0062] In step 130, processor 48 initializes user entity records 54. In some embodiments as described supra, each user entity record 54 corresponds to a given HR database record 47 an a corresponding user entity 100. When initializing user entity records 54, Additionally, when initializing user entity records 54, processor 48 can initialize user entity profiles 24 as well.

[0063] In step 132, processor 48 identifies event logs 26.

[0064] In step 134 the processor selects an unmapped event log entry 60 in a given event log 26. In embodiments herein, unmapped event log entries 60 comprise any of the event log entries no processed by steps 134-136 as described hereinbelow.

[0065] In step 136, processor 48 retrieves the selected event log entry. Upon retrieving the selected log entry, processor 48 can add a new aggregated log entry 70 to aggregated event log 52, and populate, in the new aggregated log entry, event ID 72, source 74, date 76, time 78 and event message 80 using embodiments described hereinabove.

[0066] In step 138, processor 48 identifies one or more identifiers 68 in event message 80 and stores the identified identifiers 68 to one or more extracted identifiers 84 (i.e., in one or more respective identifier information records 82).

[0067] In step 140 processor 48 normalizes the one or more extracted identifiers 84 to one or more specified formats so as to map each of the extracted identifiers to a respective account 116. In some embodiments, each account type 120 may have a corresponding specified format. Using the examples of account types described supra:

- A specified format for the account type "domain account" can be "*CompanyName[/]UserName*", where "*CompanyName*" and "*UserName*" are self-descriptive. As described supra, an example of a domain account is "*Company/jdoe*".
- A specified format for the account type "local account" can be "*ComputerID/UserName*", where "*ComputerID*" comprises an identifier for a given computing device 28 on network 32 and "*UserName*" is self-descriptive. As described supra, an example of a local account is "*host123/jdoe*".
- A specified format for the account type "cloud account" can be "*UserName[@]CompanyDomain*", where "*UserName*" *is self-descriptive* comprises an identifier for a given computing device 28 on network 32 and "*UserName*" is self-descriptive and "*CompanyDomain*" comprises a corporate domain name. As described supra, an example of a cloud account is "*john.doe@company.com*".
- A specified format for the account type "personal account" can be "*UserName[@]ProviderDomain*", where "*UserName*" *is self-descriptive* comprises and "*ProviderDomain*" comprises an email service provider domain name (e.g., Gmail™, provided by Alphabet Inc.). As described supra, an example of a personal account is "*john.doe@gmail.com*".

[0068] In some embodiments, the format for a given event is based on the source (e.g., the event log that processor 48 retrieved the event log entry corresponding to the given event, the event type, the field in the log entry corresponding to the given event) or content of the log entry corresponding to the given event. For example:

- If a given extracted identifier 84 has an email identifier format (i.e., "*local-part[@]domain*", where "*local-part*" comprises a username and "*domain*") then processor 48 can normalize the given identifier to a cloud account (e.g., "*john.doe@company.com*") or a personal account (e.g., "*john.doe@gmail.com*").
- If processor 48 extracts a given identifier 84 from a given log entry 60 from a log of a email server, and the domain is some public service, we will know it is most likely referring to a private email account (e.g., the context was that the given log entry came from a given log 26 of an email serve, and the content of the given log entry comprised a public email domain like "*@gmail*").

- SID formats can refer to local or domain accounts and are usually differentiated by content. In some embodiments, the prefix of the SID will uniquely identify the domain, or the local machine (e.g., a given computing device 28).
- GUIDs can refer to different account types, and can recognized by context (e.g., the respective types of logs 26 from which processor 48 extracted the GUIDs) or by matching the GUIDs to "ground truths" that processor 48 can extract from account database 33 (e.g., DSS™).

[0069] In some embodiments, there may be mappings from one or more extracted identifiers 84 (corresponding to respective identifiers 68) to a single account 116 (corresponding to a given account 69). For example:

- Processor 48 can map the following identifiers 84 to a given account 116 "*Company/jdoe*" whose account type 120 comprises a domain account:
  - "Company/jdoe" whose identifier type 86 comprises a domain name.
  - "*Company.com/jdoe*" whose identifier type 86 comprises a FQDN.
  - "*jdoe*" whose identifier type 86 comprises a username without any domain.
  - "S-*1-5-21-1602811402-2595058921-120187713-502*" whose identifier type 86 comprises a SID.
  - "*8c6bfd4a-4cb2-11ea-b67e-88e9fe502c1f*" whose identifier type 86 comprises a GUID.
  - "*host123\jdoe*" whose identifier type 86 comprises a local username.
  - "*john.doe@company.com*" whose identifier type 86 comprises a corporate username.

- Processor 48 can map the following identifiers 84 to a given account 116 "*host123/jdoe*" whose account type 120 comprises a local account:
  - "*jdoe*" whose identifier type 86 comprises a username without any domain.
  - "S-*1-5-21-1602811402-2595058921-120187713-502*" whose identifier type 86 comprises a SID.
  - "*8c6bfd4a-4cb2-11ea-b67e-88e9fe502c1f*" whose identifier type 86 comprises a GUID.
  - "*host123\jdoe*" whose identifier type 86 comprises a local username.

- Processor 48 can map the following identifiers 84 to a given account 116 "*john.doe@company.com*" whose account type 120 comprises a cloud account:
    - "*8c6bfd4a-4cb2-11ea-b67e-88e9fe502c1f*" whose identifier type 86 comprises a GUID.
    - "*john.doe@company.com*" whose identifier type 86 comprises a corporate username.
    - "*john.doe@gmail.com*" whose identifier type 86 comprises a personal username.

- Processor 48 can map the following identifier 84 to a given account 116 "*john.doe@gmail.com*" whose account type 120 comprises a personal account:
    - "*john.doe@gmail.com*" whose identifier type 86 comprises a personal username.

[0070] In some embodiments, processor 46 can query database records 35 to the extracted identifiers to a respective account 116.

[0071] Upon performing each mapping of a given extracted identifier 84, processor 48 stores, the mapped account (ID) 116 to mapped account 88 in the identifier information record 82 storing the given extracted identifier. If any given mapping detected is step 140 is not already stored to user entity records 54, processor 48 can add a new identifier-account mapping record in the user entity record storing the mapped account, and populate identifier 122, identifier type 124 and associated account ID 126 accordingly.

[0072] In a first normalization embodiment, can normalize a given extracted identifier 84 by string manipulation (i.e., processor 48 stores the extracted identifiers as text strings). In this embodiment, processor 48 can normalizing extracted identifiers 84 to enable correlations and queries. For example, processor 48 can use string manipulation to normalize both

- *"jdoe@company[.]onmicrosoft[.]com"*
- *"domain=company.local, username=jdoe"*

to "company\jdoe".

[0073] In a second normalization embodiment, processor 48 can normalize a given extracted identifier 84 by using domain knowledge. In this embodiment, special identifiers can indicate the type and scope of the account (e.g., at the host or main levels) mapped to the given identifier. In the following examples, processor 48 can use domain knowledge to:

- Map "*AzureAD\jdoe*" to a cloud account.

- Map "*MicrosoftAccount\jdoe*" domain to a personal Microsoft™ account.
- Map "*company\jdoe$*" to a machine account of a given host name 56 "jdoe". In this example "*$*" in the identifier indicates a machine account (i.e., "*$*" + a username).

[0074] Domain knowledge enables processor 48 to differentiate between accounts that are typically managed differently in Active Domain™ and Kerberos realms, as well as various data cloud environments.

[0075] In a third normalization embodiment, processor 48 can normalize a given extracted identifier 84 by using prior learned knowledge. In this embodiment, processor 48 can use learned roles and Directory Synchronization Service (DSS™) to determine the account for a given extracted identifier 84. In the following examples, processor 48 can use domain knowledge as follows:

- If *ad_domain_role* contains "*company*" then account type 120 is a domain account.
- If *internal_hostname_role* contains "*company*" then account type 120 is a local account
- If the event message only has a SID number, then processor 48 can pivot via the *DSS.sid* field ("*sid*" is an abbreviation for "*security identifier*" in Active Directory™) so as to map the given extracted identifier to "*company\jdoe*".
- If the given extracted identifier comprises "*john.doe@gmail[.]com*", pivot via the *DSS.upn* field ("*upn*" is an abbreviation for "*user principal name*" in Active Directory™) so as to recognize the given extracted identifier as a domain account, and then map the extracted identifier to normalized identifier "*company\jdoe*". In this example, processor 48 compares the string "*john.doe@gmail[.]com*" to the *DSS.upn* field, and if a record with that value is found, it will be considered to be a domain account, and the processor will return normalized identifier "*company\jdoe*". In some embodiments, the value in the corresponding DSS record "*DSS.netbios_domain\DSS.sam_account_name*" may comprise "company*jdoe*".

[0076] Returning to the flow diagram, in step 142, for each given mapped account 88, processor 48 associates a given user entity 100 with a given mapped account 88. In some embodiments, each user entity 100 may be associated with one or more accounts 116. For example, as described supra, the mapped accounts may comprise "*Company/jdoe*", "*host123/jdoe*", "*john.doe@company.com*"

and "*john.doe@gmail.com*". All these mapped accounts 88 may be associated with a given user entity named "*John Doe*".

[0077] In a first association embodiment, processor 46 can use information stored in HR database 46 and/or account database 33 so as to associate a given account 69 with a given user entity 67. For example, if processor 46 uses account database 33 to map a given identifier 68 to a given account 69 "*john.doe@gmail.com*", and identifies a given user entity 67 named "John Doe" in HR database 46, then the processor can associate the given account with the given user entity as they have the same name.

[0078] In a second association embodiment, processor 48 can use heuristics to associate the given user entity with the given mapped account. For example, if "*john.doe@gmail[.]com*" matches DSS display name "*John Doe*" then they likely refer to the same user entity 100.

[0079] In a third association embodiment, processor 48 can use profiling and attribution to associate the given user entity with the given mapped account. In one profiling example, processor 48 can determine that the computing device having the host name "*host_123*" is mostly used by a single user entity 100 "company\jdoe". In a second profiling example, processor 48 can determine that the account "*john.doe@gmail[.]com*" always originates log enties 60 from the computing device having the host name "*host_123*".

[0080] In a first attribution example, processor 48 can determine that the computing device having the host name "*host_123*" is a personal endpoint used by the user entity "*jdoe*". In a second attribution example, processor 48 can determine that "*john.doe@gmail[.]com*" is the personal email of the user entity "*jdoe*". In a third attribution example, processor 48 can determine that the user entity "*jdoe*" likely has access to the account "*host_123\Administrator*".

[0081] Returning to the flow diagram, in step 144, processor 48 identifies one or more of the user entities that participated in the event corresponding to the selected log entry.

[0082] In step 146, processor 48 updates, with the event indicated by the event message in the selected log entry, the user entity profile for each of the user entities identified in step 144. . In some embodiments, processor 48 can update user entity profiles 24 with the event indicated in the selected log entry only if the event was within a specified time period (e.g., the last 30 days).

[0083] In step 148, if there are any unmapped log entries 60, then the method continues with step 132. The method ends when there are no unmapped log entries 60.

[0084] Once processor 48 creates profiles 24, the processor can use the profiles to detect a single user entity 100 using multiple identifiers 122 to perform malicious activity in computing facility 20. For example, Processor 48 can:

1. Detect a cloud account "*jdoe@company[.]com*" downloaded a file "*confidential.pdf*" from Google Drive™ (provided by Alphabet Inc., Mountain View, California).
2. Detect that domain account "*Company\jdoe*" renamed file "*confidential.pdf*" to "*obscure.zip*".
3. Detect an email sent to personal email "*john.doe@gmail[.]com*" with an attachment named obscure.zip

[0085] While each of these individual events may seem legitimate, embodiments of the present invention enable correlating these three events to a single user entity 100 "John Doe". Correlating multiple events having multiple identifiers 122 enables processor 48 to detect a suspicious sequence of events that are tied to a single user entity 100.

[0086] Figure 7 is a flow diagram that schematically illustrates a method of using user entity activity profiles 24 to detect suspicious activity, in accordance with an embodiment of the present invention.

[0087] In step 150, at a time subsequent to generating profiles 24 as described in the description referencing Figure 6 hereinabove, processor 48 collects, from logs 26, a set of additional event log entries 60. In some embodiments, processor 48 can collect the additional event log entries during a specific time period (e.g., 10 minutes or a full day).

[0088] In step 152, processor 48 associates each of the events in the event messages in the additional event log entries with respective user entities 100, using embodiments described in the description referencing steps 140-142 in Figure 6 hereinabove.

[0089] In step 154, processor 48 updates status information records 104 with any updates to HR database 46 and updates user entity profiles 24 accordingly. For example, the user entity "*John Doe*" may be on vacation.

[0090] In step 154 processor 48 selects an unselected user entity 100.

[0091] In step 156, processor 48 compares the additional events for the selected user entity to user entity profile 24 of the selected user entity.

[0092] In step 158, processor 48 determines, based on the user entity profile, whether or not the additional events comprise suspicious activity. In some embodiments each user profile 24 can include information from status records 104 for the corresponding user entity 100. For example, if a given status for 114 for a given user entity 100 indicates that the given user entity is retired, and processor 48 detects events associated with the user entity subsequent to the retirement, then the processor can classify those events as suspicious since the events are not in accordance with the retirement status in the user entity profile.

[0093] If the additional events comprise suspicious activity, then in step 160, processor 48 issues an alert for the selected user entity. In one embodiment, the suspicious activity may combine a first event in a first given event log entry 60 that processor 48 used to generate the user entity profile, and a second event in a second given event log entry 60 that processor 48 collected in step 150. In this embodiment, the first and the second given event log entries mapped to different identifiers 122 associated with the same user entity 100.

[0094] To issue the alert processor 48 can perform operations such as transmitting a message to a system administrator (not shown) or restricting access to any of the accounts associated with the selected user entity.

[0095] In step 160, processor 48 updates the user entity profile of the selected user entity with the additional events associated with the selected user entity.

[0096] In step 164, if there are any unselected user entities 100 (i.e., in step 156), then the method continues with step 156. If there are no unselected user entities 100, then the method ends.

[0097] Returning to step 158, if processor 48 did not detect, based on the user entity profile, any suspicious activity in the additional events, then the method continues with step 162.

[0098] It will be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and subcombinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.

2022370400 22 Aug 2024

CLAIMS

1. A method for protecting a computer system, comprising:

identifying, by a processor, multiple user identifiers associated with a single user entity;

detecting status of the user entity, based on an information record of a first one of the user identifiers, wherein the status has a timespan;

detecting an event carried out during the timespan using a second one of the user identifiers that is not in accordance with the status of the user entity based on the information record of the first one of the user identifiers; and

issuing an alert in response to detecting the event carried out during the timespan.

2. The method according to claim 1, wherein identifying the multiple user identifiers associated with the single user entity comprises collecting a set of events, extracting respective user identifiers from the events in the set, mapping the extracted user identifiers to respective accounts, and associating the accounts with respective user entities, wherein the single user entity comprises one of the multiple user entities.

3. The method according to claim 2, wherein mapping a given extracted user identifier to a given account comprises normalizing the given user entity to a specific format, wherein the given account comprises the normalized user entity.

4. The method according to claim 2, wherein the single user entity is associated with one or more accounts.

5. The method according to claim 2, wherein multiple user identifiers map to a given account for the single user entity.

6. The method according to any of claims 1-5, wherein detecting the status comprises detecting a first event indicative of the status on a first networked entity, and wherein detecting the event comprises detecting a second event on a second networked entity different from the first networked entity.

7. The method according to any of claims 1-5, wherein detecting the status comprises detecting multiple first events during a first time period, and generating a profile in response to the multiple first events, wherein detecting the event comprises detecting one or more second events in a second time period subsequent to the first time period.

2022370400 22 Aug 2024

8. The method according to claim 1, wherein detecting the status comprises detecting, based on the first one of the user identifiers, that during the timespan the user entity was no longer employed by an organization issuing the first one of the user identifiers.

9. The method according to claim 1, wherein detecting the status comprises detecting, based on the first one of the user identifiers, that during the timespan the user entity was on leave from an organization issuing the first one of the user identifiers.

10. The method according to claim 1, wherein detecting the status comprises detecting, based on the first one of the user identifiers, a first location of the user entity during the timespan, and wherein detecting the event comprises detecting an action carried out during the timespan at a second location, different from the first location, using the second one of the user identifiers.

11. The method according to claim 1, wherein detecting the status comprises detecting, based on the first one of the user identifiers, a first device used by the user entity during the timespan, and wherein detecting the event comprises detecting an action carried out on a second device, different from the first device, during the timespan using the second one of the user identifiers.

12. The method according to claim 1, wherein detecting the status comprises detecting, based on the first one of the user identifiers, a department in an organization to which the user entity belongs during the timespan, and wherein detecting the event comprises detecting a system accessed during the timespan using the second one of the user identifiers, wherein the system is not typically accessed by members of the department to which the user entity belongs.

13. The method according to claim 1, wherein detecting the status comprises detecting, based on the first one of the user identifiers, a level of privileges of the user entity during the timespan, and wherein detecting the event comprises detecting an action carried out using the second one of the user identifiers during the timespan that is not compatible with the level of privileges.

14. An apparatus for protecting a computer network, comprising:

a network interface card (NIC); and

at least one processor configured:

to identify multiple user identifiers associated with a single user entity,

to detect status of the user entity, based on an information record of a first one of the user identifiers, wherein the status has a timespan,

to detect an event carried out during the timespan using a second one of the user identifiers that is not in accordance with the status of the user entity based on the information record of the first one of the user identifiers, and

to issue an alert in response to detecting the event carried out during the timespan.

15. The apparatus according to claim 14, wherein a given processor is configured to identify the multiple user identifiers associated with the single user entity by collecting a set of events, extracting respective user identifiers from the events in the set, mapping the extracted user identifiers to respective accounts, and associating the accounts with respective user entities, wherein the single user entity comprises one of the multiple user entities.

16. The apparatus according to claim 15, wherein a given processor is configured to map a given extracted user identifier to a given account by normalizing the given user entity to a specific format, wherein the given account comprises the normalized user entity.

17. The apparatus according to claim 15, wherein the single user entity is associated with one or more accounts.

18. The apparatus according to claim 17, wherein multiple user identifiers map to a given account for the single user entity.

19. The apparatus according to any of claims 14-18, wherein a given processor is configured to detect the status by detecting a first event indicative of the status on a first networked entity of a network, and to detect the event by detecting a second event on a second networked entity different from the first networked entity.

20. The apparatus according to any of claims 14-18, wherein a given processor is configured to detect the status by detecting multiple first events during a first time period, and wherein the given processor is further configured to generate a profile in response to the multiple first events, and wherein the given processor is configured to detect the event by detecting one or more second events in a second time period subsequent to the first time period.

21. A computer software product for protecting a computing system, the product comprising a non-transitory computer-readable medium, in which program instructions are stored, which instructions, when read by a computer, cause the computer:

to identify multiple user identifiers associated with a single user entity;

to detect a status of the user entity, based on an information record of a first one of the user identifiers, wherein the status has a timespan;

to detect an event carried out during the timespan using a second one of the user identifiers that is not in accordance with the status of the user entity based on the information record of the first one of the user identifiers; and

to issue an alert in response to detecting the event carried out during the timespan.

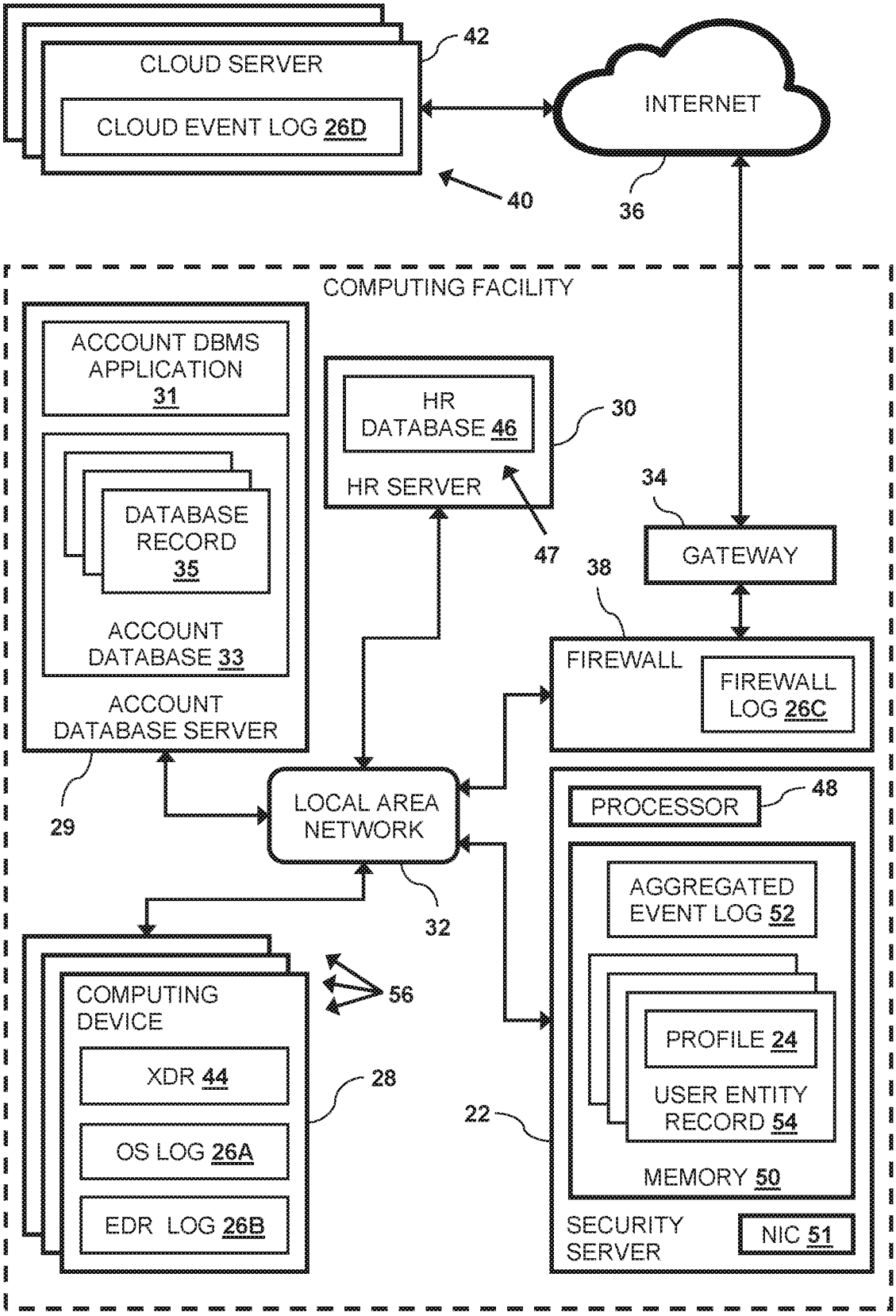
CLOUD SERVER
CLOUD EVENT LOG 26D
42
INTERNET
40
36
COMPUTING FACILITY
ACCOUNT DBMS APPLICATION 31
HR DATABASE 46
30
HR SERVER
47
34
GATEWAY
38
DATABASE RECORD 35
ACCOUNT DATABASE 33
ACCOUNT DATABASE SERVER
FIREWALL
FIREWALL LOG 26C
29
LOCAL AREA NETWORK
PROCESSOR
48
AGGREGATED EVENT LOG 52
32
56
COMPUTING DEVICE
XDR 44
28
OS LOG 26A
EDR LOG 26B
PROFILE 24
USER ENTITY RECORD 54
22
MEMORY 50
SECURITY SERVER
NIC 51

FIG. 1

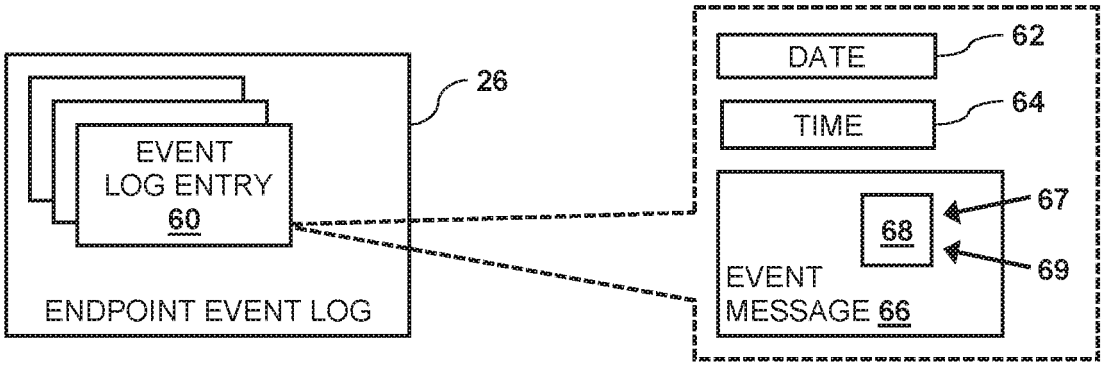
26
EVENT
LOG ENTRY
60
ENDPOINT EVENT LOG
DATE
62
TIME
64
68
67
69
EVENT
MESSAGE 66

FIG. 2

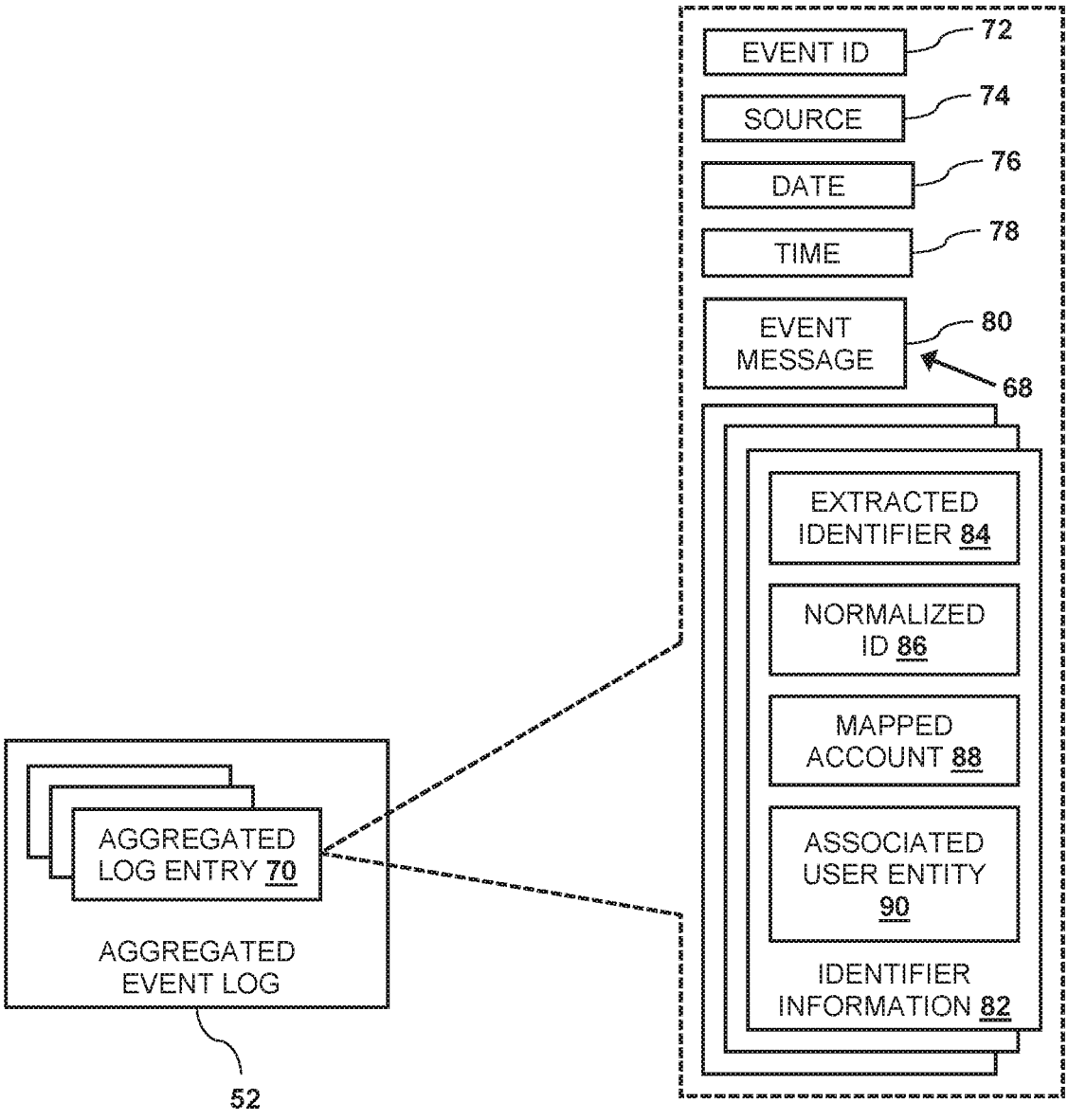
EVENT ID
72
SOURCE
74
DATE
76
TIME
78
EVENT
MESSAGE
80
68
EXTRACTED
IDENTIFIER 84
NORMALIZED
ID 86
MAPPED
ACCOUNT 88
ASSOCIATED
USER ENTITY
90
IDENTIFIER
INFORMATION 82
AGGREGATED
LOG ENTRY 70
AGGREGATED
EVENT LOG
52

FIG. 3

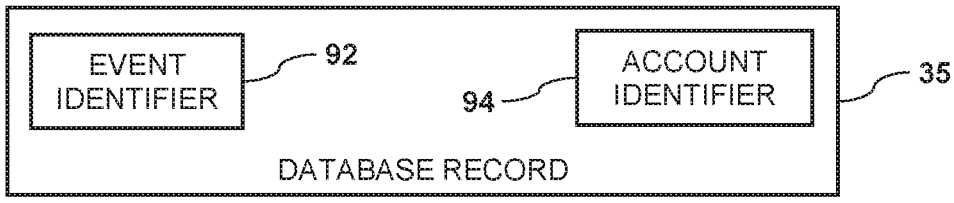
EVENT IDENTIFIER
92
94
ACCOUNT IDENTIFIER
35
DATABASE RECORD

FIG. 4

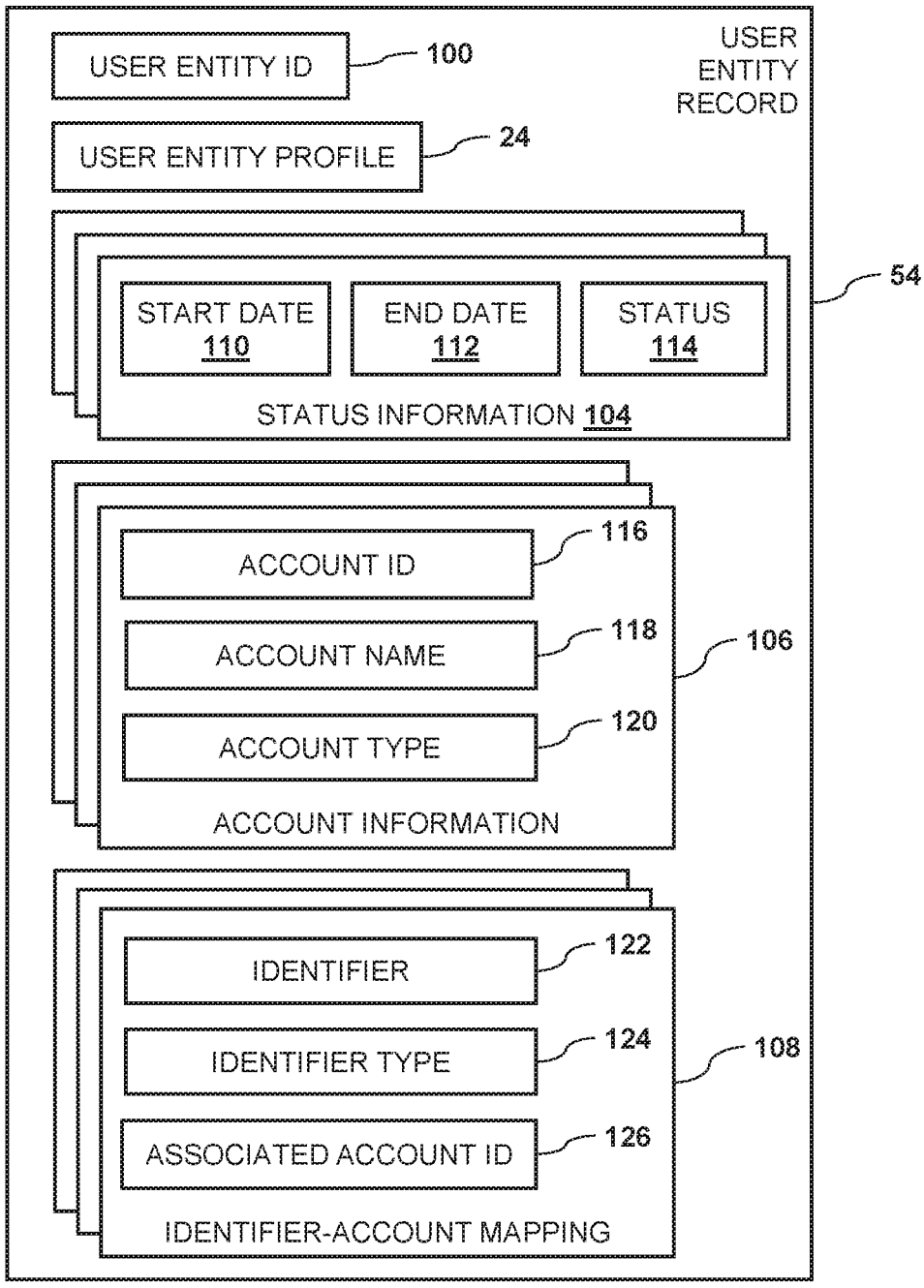
USER ENTITY ID
100
USER ENTITY RECORD
USER ENTITY PROFILE
24
START DATE 110
END DATE 112
STATUS 114
54
STATUS INFORMATION 104
ACCOUNT ID
116
ACCOUNT NAME
118
ACCOUNT TYPE
120
106
ACCOUNT INFORMATION
IDENTIFIER
122
IDENTIFIER TYPE
124
ASSOCIATED ACCOUNT ID
126
108
IDENTIFIER-ACCOUNT MAPPING

FIG. 5

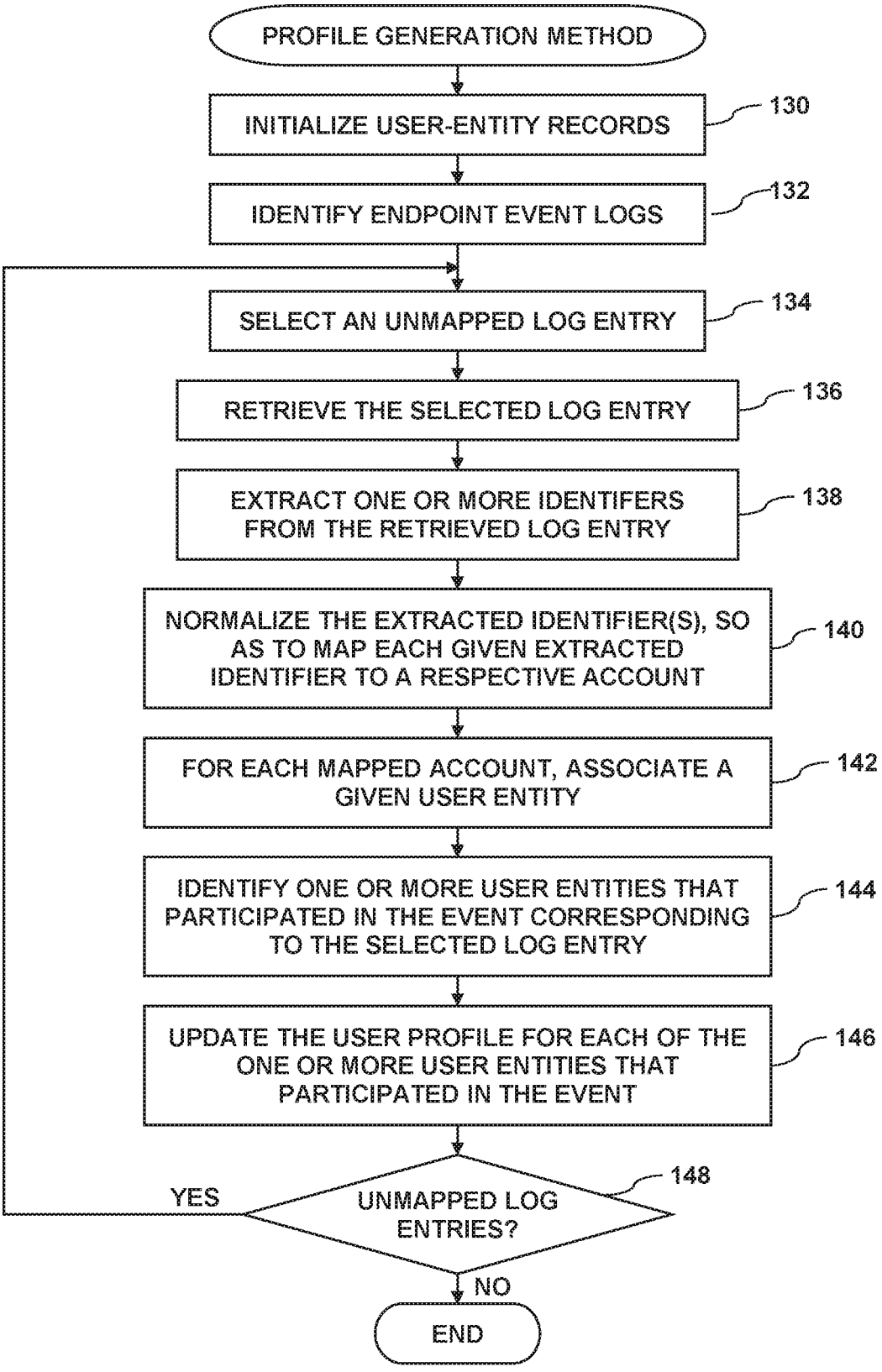
PROFILE GENERATION METHOD
INITIALIZE USER-ENTITY RECORDS
130
IDENTIFY ENDPOINT EVENT LOGS
132
SELECT AN UNMAPPED LOG ENTRY
134
RETRIEVE THE SELECTED LOG ENTRY
136
EXTRACT ONE OR MORE IDENTIFERS FROM THE RETRIEVED LOG ENTRY
138
NORMALIZE THE EXTRACTED IDENTIFIER(S), SO AS TO MAP EACH GIVEN EXTRACTED IDENTIFIER TO A RESPECTIVE ACCOUNT
140
FOR EACH MAPPED ACCOUNT, ASSOCIATE A GIVEN USER ENTITY
142
IDENTIFY ONE OR MORE USER ENTITIES THAT PARTICIPATED IN THE EVENT CORRESPONDING TO THE SELECTED LOG ENTRY
144
UPDATE THE USER PROFILE FOR EACH OF THE ONE OR MORE USER ENTITIES THAT PARTICIPATED IN THE EVENT
146
UNMAPPED LOG ENTRIES?
148
YES
NO
END

FIG. 6

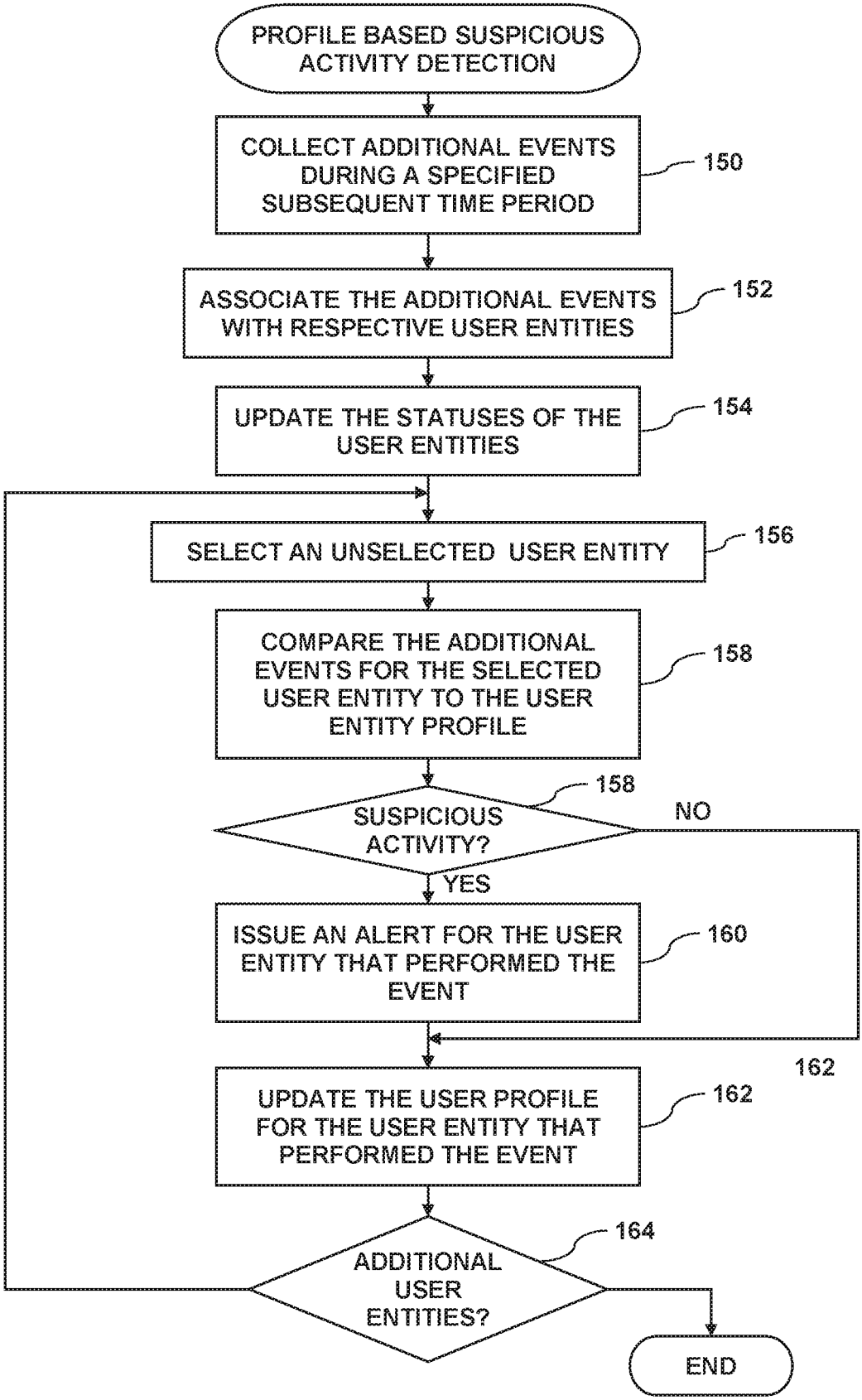
PROFILE BASED SUSPICIOUS ACTIVITY DETECTION
COLLECT ADDITIONAL EVENTS DURING A SPECIFIED SUBSEQUENT TIME PERIOD
150
ASSOCIATE THE ADDITIONAL EVENTS WITH RESPECTIVE USER ENTITIES
152
UPDATE THE STATUSES OF THE USER ENTITIES
154
SELECT AN UNSELECTED USER ENTITY
156
COMPARE THE ADDITIONAL EVENTS FOR THE SELECTED USER ENTITY TO THE USER ENTITY PROFILE
158
158
SUSPICIOUS ACTIVITY?
NO
YES
ISSUE AN ALERT FOR THE USER ENTITY THAT PERFORMED THE EVENT
160
162
UPDATE THE USER PROFILE FOR THE USER ENTITY THAT PERFORMED THE EVENT
162
164
ADDITIONAL USER ENTITIES?
END

FIG. 7