

(19) **DANMARK**

(10) **DK/EP 3528462 T3**



(12) **Oversættelse af  
europæisk patentskrift**

Patent- og  
Varemærkestyrelsen

- 
- (51) Int.Cl.: **H 04 L 9/40 (2022.01)** **G 06 F 21/55 (2013.01)** **H 04 L 67/12 (2022.01)**  
**H 04 L 51/212 (2022.01)**
- (45) Oversættelsen bekendtgjort den: **2024-02-26**
- (80) Dato for Den Europæiske Patentmyndigheds bekendtgørelse om meddelelse af patentet: **2024-01-24**
- (86) Europæisk ansøgning nr.: **19158049.7**
- (86) Europæisk indleveringsdag: **2019-02-19**
- (87) Den europæiske ansøgnings publiceringsdag: **2019-08-21**
- (30) Prioritet: **2018-02-20 US 201862632623 P**
- (84) Designerede stater: **AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**
- (73) Patenthaver: **Darktrace Holdings Limited, Maurice Wilkes Building , St John's Innovation Park , Cowley Road, Cambridge, Cambridgeshire CB4 0DS, Storbritannien**
- (72) Opfinder: **HUMPHREY, Dickon, 2 Lents Way, Cambridge, Cambridgeshire CB4 1UA, Storbritannien**  
**BISPAM, Matthew, The Abbey, Burton Road, Repton, Derbyshire DE65 6FL, Storbritannien**  
**STOCKDALE, Jack, The Old Rectory, Green End, Cambridge, Cambridgeshire CB25 9FD, Storbritannien**
- (74) Fuldmægtig i Danmark: **Zacco Denmark A/S, Arne Jacobsens Allé 15, 2300 København S, Danmark**
- (54) Benævnelse: **Fremgangsmåde til at dele en cybersikkerhedstrusselsanalyse og defensive forholdsregler i et fællesskab**
- (56) Fremdragne publikationer:  
**US-A1- 2016 261 621**  
**US-A1- 2017 118 236**  
**US-A1- 2017 262 633**  
**LI ZHOU ET AL: "Operational Security Log Analytics for Enterprise Breach Detection", 2016 IEEE CYBERSECURITY DEVELOPMENT (SECDEV), IEEE, 3 November 2016 (2016-11-03), pages 15-22, XP033058161, DOI: 10.1109/SECDEV.2016.015 [retrieved on 2017-02-01]**



# DESCRIPTION

## FIELD

**[0001]** Embodiments of the design provided herein generally relate to a cyber threat defense system. In an embodiment, a cyber threat defense system may use previously discovered cyber threats to inoculate against future threats.

## BACKGROUND

**[0002]** In the cyber security environment, firewalls, endpoint security methods and other tools such as security information and event management systems (SIEMs) and restricted environments, such as sandboxes, are deployed to enforce specific policies and provide protection against certain threats. These tools currently form an important part of an organization's cyber defense strategy, but they are insufficient in the new age of cyber threat.

**[0003]** Cyber threat, including email threats, viruses, trojan horses, and worms, can subtly and rapidly cause harm to a network. Additionally, human users may wreak further damage to the system by malicious action. A cyber security system has to identify each of these cyber threats as they evolve.

**[0004]** US patent application with publication number US2017/0118236 discloses a system for detecting security threats in a local network. A security analytics system collects data about entities in the local network. The security analytics system parses the raw data into data fields and identifies a subset of the data fields. The security analytics system identifies relationships between the plurality of entities, generates a set of features between the plurality of entities, generates a set of features based on the structured data and the identified relationships, and generates one or more threat scores based on the set of features. The security analytics system detects malicious behaviour performed by an entity in the local network based on the generated threat scores.

**[0005]** US patent application with publication number US2017/0262633 discloses systems and methods for automated machine-learning, zero-day malware detection. Embodiments include a system and method for detecting malware using multi-stage file-typing and, optionally, pre-processing, with fall-through options.

**[0006]** US patent application with publication number US2016/0261621 discloses a device which may receive behaviour information that identifies a first user, of a first set of users, in association with a behaviour. The behaviour may relate to one or more requests, from a client device being used by the first user, to access a network resource. The device may determine, based on a model, whether the behaviour is normal.

**SUMMARY**

**[0007]** A cyber threat defense system can leverage identifying threats by spotting deviations from normal behavior to create a system-wide inoculation regimen. The cyber threat defense system can have a comparison module to execute a comparison of input data for a network entity to at least one machine-learning model of a generic network entity using a normal behavior benchmark to spot behavior deviating from normal benign behavior. The comparison module can identify whether the network entity is in a breach state. The cyber threat defense system can have a cyber threat module to identify whether the breach state and a chain of relevant behavioral parameters correspond to a cyber threat. The cyber threat defense system can have an inoculation module to send an inoculation notice to warn of a potential cyber threat to a target device.

**[0008]** These and other features of the design provided herein can be better understood with reference to the drawings, description, and claims, all of which form the disclosure of this patent application.

**[0009]** Aspects of the present invention are defined by the independent claims below to which reference should now be made. Optional features are defined by the dependent claims.

**DRAWINGS**

**[0010]** The drawings refer to some embodiments of the design provided herein in which:

Figure 1 illustrates a block diagram of an embodiment of a cyber threat defense system with a cyber threat module that references machine-learning models that are trained on the normal behavior of a network entity to identify cyber threats by identifying deviations from normal behavior.

Figure 2 illustrates a block diagram of an embodiment of an example chain of unusual behavior for the email(s) in connection with the rest of the network under analysis.

Figure 3 illustrates an example cyber threat defense system protecting an example network.

Figure 4 illustrates in a block diagram the integration of the threat detection system with other network protections.

Figure 5 illustrates an application of a cyber threat defense system using advanced machine-learning to detect anomalous behavior.

Figure 6 illustrates a flowchart of an embodiment of a method for modeling human activity, machine activity, or other activity.

Figure 7 illustrates an inoculation component of the cyber threat defense system.

Figure 8 illustrates a flowchart of an embodiment of a method for developing an inoculation record based on the detection of an anomalous event.

Figure 9 illustrates a block diagram of a threat risk parameter.

Figure 10 illustrates a flowchart of an embodiment of a method for generating a threat risk parameter.

Figure 11 illustrates a block diagram of an inoculation record.

Figure 12 illustrates a flowchart of an embodiment of a method for generating an inoculation record.

Figure 13 illustrates a flowchart of an embodiment of a method for sending an inoculation notice.

Figure 14 illustrates a block diagram of a benchmark matrix.

Figure 15 illustrates a flowchart of an embodiment of a method for comparing analyzed input data to a benchmark to trigger a notice.

Figure 16 illustrates a flowchart of an embodiment of a method for sending an inoculation report.

Figure 17 illustrates an example network to be protected by the cyber threat defense system.

**[0011]** While the design is subject to various modifications, equivalents, and alternative forms, specific embodiments thereof have been shown by way of example in the drawings and will now be described in detail. It should be understood that the design is not limited to the particular embodiments disclosed, but - on the contrary - the intention is to cover all modifications, equivalents, and alternative forms using the specific embodiments.

## **DESCRIPTION**

**[0012]** In the following description, numerous specific details are set forth, such as examples of specific data signals, named components, number of servers in a system, etc., in order to provide a thorough understanding of the present design. It will be apparent, however, to one of ordinary skill in the art that the present design can be practiced without these specific details. In other instances, well known components or methods have not been described in detail but rather in a block diagram in order to avoid unnecessarily obscuring the present design. Further, specific numeric references such as a first server, can be made. However, the specific numeric reference should not be interpreted as a literal sequential order but rather interpreted

that the first server is different than a second server. Thus, the specific details set forth are merely exemplary. Also, the features implemented in one embodiment may be implemented in another embodiment where logically possible. The specific details can be varied from and still be contemplated to be within the scope of the present design. The term coupled is defined as meaning connected either directly to the component or indirectly to the component through another component.

**[0013]** In general, the cyber threat defense system may use artificial intelligence to analyze cyber security threats. Figure 1 illustrates a block diagram of an embodiment of a cyber threat defense system with a cyber threat module that references machine-learning models that are trained on the normal behavior of network activity and user activity associated with a network. The cyber threat module determines a threat risk parameter that factors in 'what is a likelihood of a chain of one or more unusual behaviors of email activity, network activity, and user activity under analysis that fall outside of being a normal benign behavior;' and thus, are likely malicious behavior.

**[0014]** The cyber threat defense system 100 may protect against cyber security threats from an e-mail system or other communication system, as well as its network. The cyber threat defense system 100 may include components such as i) a trigger module, ii) a gather module, iii) a data store, iv) a network module, v) an email module, vi) a coordinator module, vii) a comparison module, viii) a cyber threat module, ix) a user interface module, x) a researcher module, xi) an autonomous response module, xii) an inoculation module, xiii) a communication module, xiv) at least one input or output (I/O) port to securely connect to other ports as required, xv) one or more machine-learning models such as a first Artificial Intelligence model trained on characteristics of vectors for malicious activity and related data, a second Artificial Intelligence model trained on potential cyber threats, and one or more Artificial Intelligence models each trained on different users, devices, system activities and interactions between entities in the system, and other aspects of the system, as well as xiv) other similar components in the cyber threat defense system.

**[0015]** A trigger module may detect time stamped data indicating one or more i) events and/or ii) alerts from I) unusual or II) suspicious behavior/activity are occurring and then triggers that something unusual is happening. Accordingly, the gather module is triggered by specific events and/or alerts of i) an abnormal behavior, ii) a suspicious activity, and iii) any combination of both. The inline data may be gathered on the deployment from a data store when the traffic is observed. The scope and wide variation of data available in this location results in good quality data for analysis. The collected data is passed to the comparison module and the cyber threat module.

**[0016]** The gather module may comprise of multiple automatic data gatherers that each look at different aspects of the data depending on the particular hypothesis formed for the analyzed event and/or alert. The data relevant to each type of possible hypothesis will be automatically pulled from additional external and internal sources. Some data is pulled or retrieved by the gather module for each possible hypothesis. A feedback loop of cooperation occurs between

the gather module, the email module monitoring email activity, the network module monitoring network activity, the comparison module to apply one or more models trained on different aspects of this process, and the cyber threat module to identify cyber threats based on comparisons by the comparison module. While an email module is mentioned, a similar module may be applied to other communication systems, such as text messaging and other possible vectors for malicious activity. Each hypothesis of typical threats can have various supporting points of data and other metrics associated with that possible threat, such as a human user insider attack, inappropriate network behavior, or email behavior or malicious software or malware attack, inappropriate network behavior, or email behavior. A machine-learning algorithm will look at the relevant points of data to support or refute that particular hypothesis of what the suspicious activity or abnormal behavior related for each hypothesis on what the suspicious activity or abnormal behavior relates to. Networks have a wealth of data and metrics that may be collected. The gatherers may then filter or condense the mass of data down into the important or salient features of data.

**[0017]** The cyber threat module may also use one or more machine-learning models trained on cyber threats in the network. The cyber threat module may reference the models that are trained on the normal behavior of email activity and user activity associated with the network. The cyber threat module can reference these various trained machine-learning models and data from the network module, the email module, and the trigger module. The cyber threat module can determine a threat risk parameter that factors in how the chain of unusual behaviors correlate to potential cyber threats and 'what is a likelihood of this chain of one or more unusual behaviors of the network activity and user activity under analysis that fall outside of being a normal benign behavior;' and thus, is malicious behavior.

**[0018]** The one or more machine-learning models can be self-learning models using unsupervised learning and trained on a normal behavior of different aspects of the system. For example, these aspects may include email activity and user activity associated with an email system, network activity and user activity associated with the network, and other aspects. The self-learning models of normal behavior are regularly updated. The self-learning model of normal behavior is updated when new input data is received that is deemed within the limits of normal behavior. A normal behavior threshold is used by the model as a moving benchmark of parameters that correspond to a normal pattern of life for the computing system. The normal behavior threshold is varied according to the updated changes in the computer system allowing the model to spot behavior on the computing system that falls outside the parameters set by the moving benchmark.

**[0019]** The comparison module can compare the analyzed metrics on the user activity and network activity compared to their respective moving benchmark of parameters that correspond to the normal pattern of life for the computing system used by the self-learning machine-learning models and the corresponding potential cyber threats.

**[0020]** The comparison module is configured to execute a comparison of input data to at least one machine-learning model to spot behavior on the network deviating from a normal benign

behavior of that network entity. The comparison module receives the input data via one or more ports from one or more probes. The at least one machine-learning model is trained on a generic normal benign behavior of a generic network entity. The at least one machine uses a normal behavior benchmark describing parameters corresponding to a normal pattern of activity for that network entity. The comparison module can use the comparison to identify whether the network entity is in a breach state of the normal behavior benchmark.

**[0021]** The comparison module can be integrated with the cyber threat module. The cyber threat defense system 100 may also include one or more machine-learning models trained on gaining an understanding of a plurality of characteristics on a transmission and related data including classifying the properties of the transmission and its meta data. The cyber threat module can then determine, in accordance with the analyzed metrics and the moving benchmark of what is considered normal behavior, a cyber-threat risk parameter indicative of a likelihood of a cyber-threat.

**[0022]** The cyber threat module can also reference the machine-learning models trained on a transmission and related data to determine if a transmission or a series of transmissions under analysis have potentially malicious characteristics. The cyber threat module can also factor this transmission characteristics analysis into its determination of the threat risk parameter. The cyber threat module can generate a set of incident data describing an anomalous event by an entity, here representing a user, a device participating in the network or a number of devices or users on a network behaving in the same anomalous manner. The cyber threat module can use the incident data to determine whether the anomalous event indicates a breach state representing a malicious incident or confidential data exposure. To do this, the cyber threat module can use the user interface module to present the incident data to a user analyst for review. The user interface module is configured to present a description of the breach state to a user analyst. If the user analyst determines that the description of the breach state describes a cyber threat, the user analyst can instruct the user interface module to create an inoculation notice based on the observed breach state to be transmitted to all subscribed devices. If the user analyst determines that the description of the breach state does not describe a cyber threat, the user interface module will not transmit an inoculation notice to other subscribed devices.

**[0023]** Alternately, the cyber threat module can execute an autonomous analyst to use machine-learning to determine whether the network entity in the breach state is a cyber threat. The cyber threat module is configured to identify whether the breach state identified by the comparison module and a chain of relevant behavioral parameters deviating from the normal benign behavior of that network entity correspond to a cyber threat.

**[0024]** The cyber threat defense system 100 may use multiple machine-learning models. Each machine-learning model may be trained on specific aspects of the normal pattern of life for the system such as devices, users, network traffic flow, outputs from one or more cyber security analysis tools analyzing the system, and others. One or more machine-learning models may also be trained on characteristics and aspects of all manner of types of cyber threats. One or

more machine-learning models may also be trained by observing vectors for delivery of malware, such as network activity or emails themselves.

**[0025]** The email module monitoring email activity and the network module monitoring network activity may both feed their data to a coordinator module to correlate causal links between these activities to supply this input into the cyber threat module.

**[0026]** The cyber threat module can also factor this network activity link to a particular email causal link analysis into its determination of the threat risk parameter. The cyber threat defense system 100 uses various probes to collect the user activity, such as network activity and email activity, and then feed that activity to the data store, as well as the cyber threat module and the machine-learning models. The cyber threat module uses the collected data to draw an understanding of the email activity and user activity in the network as well as updates a training for the one or more machine-learning models trained on this network and its users. For example, email traffic can be collected by putting hooks into the e-mail application, such as Outlook or Gmail, and/or monitoring the internet gateway from which the emails are routed through. Additionally, probes may collect network data and metrics via one of the following methods: port spanning the organizations existing network equipment; inserting or re-using an in-line network tap, and/or accessing any existing repositories of network data.

**[0027]** The cyber threat defense system 100 may supplement the data provided to the users and cyber professionals using a researcher module. The researcher module operates an artificial intelligence (AI) algorithm to assess whether the anomalous network activity has previously appeared in other published threat research or known lists of malicious files or internet addresses. The researcher module can consult internal threat databases or external public sources of threat data. The researcher module can collect an outside data set describing at least one of an action or a state related to the cyber threat present outside of the network from at least one data source outside the network.

**[0028]** The cyber threat defense system 100 can then take actions to counter detected potential cyber threats. The autonomous response module, rather than a human taking an action, can be configured to cause one or more rapid autonomous actions to be taken to contain the cyber threat when the threat risk parameter from the cyber threat module is equal to or above an actionable threshold. The cyber threat module's configured cooperation with the autonomous response module, to cause one or more autonomous actions to be taken to contain the cyber threat, improves computing devices in the email system by limiting an impact of the cyber threat from consuming unauthorized CPU cycles, memory space, and power consumption in the computing devices via responding to the cyber threat without waiting for some human intervention.

**[0029]** The cyber threat defense system 100 may have an inoculation module to communicate with other target devices both inside and outside the network. The inoculation module may use incident data describing a breach state by a user or device, acting as a network entity, to warn other computing devices of potential cyber threats. The inoculation module is configured to

generate an inoculation pattern describing the breach state and the chain of relevant behavioral parameters corresponding to the cyber threat identified by the cyber threat module, using the identified cyber threat as a template for developing an inoculation pattern. The inoculation module can anonymize the inoculation pattern to remove any personally identifiable information for the network entity from the inoculation pattern.

**[0030]** The inoculation module is configured to generate an inoculation notice having the inoculation pattern to warn a target device of a potential breach state of normal behavior corresponding to that cyber threat by the target device. The inoculation module is configured to use the communication module to send the inoculation notice to a device, referred to here as a target device, on a network protected by an affiliated cyber threat defense system via at least one output port.

**[0031]** The communication module can broadcast the inoculation notice to every device on the network. Alternately, the communication module can select a target device for notification of inoculation using clustering to identify commonalities between a network entity and other devices on the network. The inoculation module is configured to create an entity cluster to group the network entity with other entities of the network based on the chain of relevant behavior parameters of the inoculation pattern. The communication module is configured to select a target device for notification regarding the inoculation pattern based on the entity cluster.

**[0032]** The inoculation module to generate an inoculation record to store the inoculation pattern. The inoculation module can store the inoculation record in a network-accessible inoculation database in the data store. The inoculation module can generate a remediation action instruction to describe at least one action to remediate the breach state. If the cybersecurity appliance possesses an autonomous response module, a user analyst can enter a remediation action instruction using the user interface module. Alternately, if the cybersecurity appliance possesses an autonomous response module, the inoculation module can automatically generate the remediation action instruction based on a similar cyber threat. If the cybersecurity appliance does not possess an autonomous response module, the remediation action can be made as a passive recommendation. The inoculation record can associate the inoculation pattern with a context data set describing at least one of a network entity action and a network entity state related to the cyber threat identified by the cyber threat module. The inoculation record associates the inoculation pattern with an outside data set collected from at least one data source outside the network describing at least one of an outside action and an outside state related to the cyber threat. The inoculation module is configured to update the inoculation pattern in the inoculation record based on a subsequent event that conforms with the inoculation pattern.

**[0033]** The inoculation module can generate an inoculation report listing any inoculation records stored in a network-accessible inoculation database at the data store. The inoculation module can send the inoculation report via at least one output port to a target device to instruct the target device to retrieve an inoculation record from the network-accessible inoculation

database. The target device may retrieve the inoculation record from the network-accessible inoculation database via at least one input port. As with the inoculation notice, the communication module can broadcast the inoculation report to every device on the network. Alternately, the communication module can select a target device for an inoculation report using clustering to identify commonalities between a network entity and other devices on the network.

**[0034]** The cyber threat module can generate a threat risk parameter listing a set of values describing aspects of the breach state. The cyber threat module can populate the threat risk parameter with a confidence score, a severity score, and a consequence score. The confidence score indicates a threat likelihood describing a probability that the breach state is a cyber threat. The severity score indicates a percentage that the network entity in the breach state is deviating from the at least one model. The consequence score indicates a severity of damage attributable to the cyber threat.

**[0035]** The inoculation module can generate a benchmark matrix having a set of benchmark scores to determine whether to send the inoculation notice. The inoculation module may adjust the benchmark scores based on the collected data for the network entity. The inoculation module may assign a weight to each benchmark score to assign a relative importance to each benchmark score. The inoculation module may compare the threat risk parameter to a benchmark matrix having a set of benchmark scores to determine whether to send an inoculation notice.

**[0036]** The cyber threat defense system 100 may be hosted on a device, on one or more servers, or in its own cyber threat appliance platform.

**[0037]** Figure 2 illustrates a block diagram of an embodiment of an example chain of unusual behavior for the email(s) in connection with the rest of the network under analysis.

**[0038]** The user interface can display a graph 200 of an example chain of unusual behavior for the email(s) in connection with the rest of the network under analysis.

**[0039]** The network & email module can tie the alerts and events from the email realm to the alerts and events from the network realm.

**[0040]** The cyber threat module cooperates with one or more machine-learning models. The one or more machine-learning models are trained and otherwise configured with mathematical algorithms to infer, for the cyber threat analysis, 'what is possibly happening with the chain of distinct alerts and/or events, which came from the unusual pattern,' and then assign a threat risk associated with that distinct item of the chain of alerts and/or events forming the unusual pattern.

**[0041]** This is 'a behavioral pattern analysis' of what are the unusual behaviors of the network entity, such as a network, a system, a device, a user, or an email, under analysis by the cyber

threat module and the machine-learning models. The cyber defense system uses unusual behavior deviating from the normal behavior and then builds a chain of unusual behavior and the causal links between the chain of unusual behavior to detect cyber threats. An example behavioral pattern analysis of what are the unusual behaviors may be as follows. The unusual pattern may be determined by filtering out what activities, events, or alerts that fall within the window of what is the normal pattern of life for that network entity under analysis. Then the pattern of the behavior of the activities, events, or alerts that are left, after the filtering, can be analyzed to determine whether that pattern is indicative of a behavior of a malicious actor, such as a human, a program, an email, or other threat. The defense system can go back and pull in some of the filtered out normal activities to help support or refute a possible hypothesis of whether that pattern is indicative of a behavior of a malicious actor. An example behavioral pattern included in the chain is shown in the graph over a time frame of, an example, 7 days. The defense system detects a chain of anomalous behavior of unusual data transfers three times, unusual characteristics in network actions in the monitored system three times which seem to have some causal link to the unusual data transfers. Likewise, twice unusual credentials tried unusual behavior of trying to access to sensitive areas or malicious IP addresses and the user associated with the unusual credentials trying unusual behavior has a causal link to at least one of those three emails with unusual characteristics. When the behavioral pattern analysis of any individual behavior or of the chain as a group is believed to be indicative of a malicious threat, then a score of how confident the defense system is in this assessment of identifying whether the unusual pattern was caused by a malicious actor is created. Next, also assigned is a threat level parameter (e.g. score or probability) indicative of what level of threat does this malicious actor pose to the system. Lastly, the cyber threat defense system is configurable in its user interface of the defense system on what type of automatic response actions, if any, the defense system may take when for different types of cyber threats that are equal to or above a configurable level of threat posed by this malicious actor.

**[0042]** The cyber threat module may chain the individual alerts and events that form the unusual pattern into a distinct item for cyber threat analysis of that chain of distinct alerts or events. The cyber threat module may reference the one or more machine-learning models trained on e-mail threats to identify similar characteristics from the individual alerts or events forming the distinct item made up of the chain of alerts or events forming the unusual pattern.

**[0043]** One or more machine-learning models may also be trained on characteristics and aspects of all manner of types of cyber threats to analyze the threat risk associated with the chain or cluster of alerts or events forming the unusual pattern. The machine-learning technology, using advanced mathematics, can detect previously unidentified threats, without relying on prescribed rules, and automatically defend networks.

**[0044]** The models may perform by the threat detection through a probabilistic change in normal behavior through the application of an unsupervised Bayesian mathematical model to detect behavioral change in computers and computer networks. The core threat detection system is termed the 'Bayesian probabilistic'. The Bayesian probabilistic approach can

determine periodicity in multiple time series data and identify changes across single and multiple time series data for the purpose of anomalous behavior detection. From the email and network raw sources of data, a large number of metrics can be derived each producing time series data for the given metric.

**[0045]** The detectors in the cyber threat module including the network module and email module components can be discrete mathematical models that implement a specific mathematical method against different sets of variables with the target. Thus, each model is specifically targeted on the pattern of life of alerts and/or events coming from, for example, i) that cyber security analysis tool, ii) analyzing various aspects of the emails, iii) coming from specific devices and/or users within a system, etc.

**[0046]** At its core, the cyber threat defense system mathematically characterizes what constitutes 'normal' behavior based on the analysis of a large number/set of different measures of a devices network behavior. The cyber threat defense system can build a sophisticated 'pattern of life' - that understands what represents normality for every person, device, email activity, and network activity in the system being protected by the cyber threat defense system.

**[0047]** As discussed, each machine-learning model may be trained on specific aspects of the normal pattern of life for the system such as devices, users, network traffic flow, outputs from one or more cyber security analysis tools analyzing the system, email contact associations for each user, email characteristics, and others. The one or more machine-learning models may use at least unsupervised learning algorithms to establish what is the normal pattern of life for the system. The machine-learning models can train on both i) the historical normal distribution of alerts and events for that system and ii) a normal distribution information from similar peer systems to establish the normal pattern of life of the behavior of alerts or events for that system. Another set of machine-learning models train on characteristics of emails and the activities and behavior of its email users to establish a normal for these.

**[0048]** The models can leverage at least two different approaches to detecting anomalies: such as comparing each system's behavior to its own history and comparing that system to its peers' history or such as comparing an email to both characteristics of emails and the activities and behavior of its email users. This multiple source comparison allows the models to avoid learning existing bad behavior as 'a normal behavior', because compromised entities, such as devices, users, components, emails will exhibit behavior different to their immediate peers.

**[0049]** In addition, the one or more machine-learning models can use the comparison of i) the normal pattern of life for that system corresponding to the historical normal distribution of alerts and events for that system mapped out in the same multiple dimension space to ii) the current chain of individual alerts and events behavior under analysis. This comparison can yield detection of the one or more unusual patterns of behavior within the plotted individual alerts or events, which allows the detection of previously unidentified cyber threats compared to finding cyber threats with merely predefined descriptive objects or signatures. Thus, increasingly

intelligent malicious cyber threats, picking and choosing when they take their actions in order to generate low level alerts and event, will still be detected, even though they have not yet been identified by other methods of cyber analysis. These intelligent malicious cyber threats can include malware, spyware, key loggers, malicious links in an email, malicious attachments in an email, and others as well as nefarious internal information technology staff who know intimately how to not set off any high-level alerts or events.

**[0050]** The plotting and comparison are a way to filter out what is normal for that system and then be able to focus the analysis on what is abnormal or unusual for that system. Then for each hypothesis of what could be happening with the chain of unusual events or alerts, the gather module may gather additional metrics from the data store including the pool of metrics originally considered 'normal behavior' to support or refute each possible hypothesis of what could be happening with this chain of unusual behavior under analysis.

**[0051]** Note, each of the individual alerts or events in a chain of alerts or events that form the unusual pattern can indicate subtle abnormal behavior. Thus, each alert or event can have a low threat risk associated with that individual alert or event. However, when analyzed as a distinct chain or grouping of alerts or events behavior forming the chain of unusual pattern by the one or more machine-learning models, that distinct chain of alerts or events can be determined to now have a much higher threat risk than any of the individual and/or events in the chain.

**[0052]** In addition, modern cyber-attacks can be of such severity and speed that a human response cannot happen quickly enough. Thanks to these self-learning advances, a machine may uncover these emerging threats and deploy appropriate, real-time responses to fight back against the most serious cyber threats.

**[0053]** The threat detection system has the ability to self-learn and detect normality in order to spot true anomalies, allowing organizations of all sizes to understand the behavior of users and machines on their networks at both an individual and group level. Monitoring behaviors, rather than using predefined descriptive objects and/or signatures, means that more attacks can be spotted ahead of time and extremely subtle indicators of wrongdoing can be detected. Unlike traditional legacy defenses, a specific attack type or new malware does not have to have been seen first before it can be detected. A behavioral defense approach mathematically models both machine, email, and human activity behaviorally, at and after the point of compromise, in order to predict and catch today's increasingly sophisticated cyber-attack vectors. It is thus possible to computationally establish what is normal, in order to then detect what is abnormal. In addition, the machine-learning constantly revisits assumptions about behavior, using probabilistic mathematics. The cyber threat defense system's unsupervised machine-learning methods do not require training data with pre-defined labels. Instead, unsupervised machine-learning methods may identify key patterns and trends in the data, without the need for human input.

**[0054]** The user interface and output module may also project the individual alerts and/or

events forming the chain of behavior onto the user interface with at least three-dimensions of i) a horizontal axis of a window of time, ii) a vertical axis of a scale indicative of the threat risk assigned for each alert and/or event in the chain and a third dimension of iii) a different color for the similar characteristics shared among the individual alerts and events forming the distinct item of the chain. The different color may be red, blue, yellow, or others. For gray scale, the user interface may use different shades of gray, black, and white with potentially different hashing patterns. These similarities of events or alerts in the chain may be, for example, alerts or events that are coming from same device, same user credentials, same group, same source identifiers, same destination Internet Protocol addresses, same types of data transfers, same type of unusual activity, same type of alerts, same rare connection being made, same type of events, or others, so that a human can visually see what spatially and content-wise is making up a particular chain rather than merely viewing a textual log of data. Note, once the human mind visually sees the projected pattern and corresponding data, then the human can ultimately decide if a cyber threat is posed. Again, the at least three-dimensional projection helps a human synthesize this information more easily. The visualization onto the user interface allows a human to see data that supports or refutes why the cyber threat defense system thinks these aggregated alerts or events could be potentially malicious. Also, instead of generating the simple binary outputs 'malicious' or 'benign,' the cyber threat defense system's mathematical algorithms produce outputs that indicate differing degrees of potential compromise.

### **Defense System**

**[0055]** Figure 3 illustrates an example cyber threat defense system protecting an example network. The example network Figure 3 illustrates a network of computer systems 50 using a threat detection system. The system depicted by Figure 3 is a simplified illustration, which is provided for ease of explanation of the invention. The system 50 comprises a first computer system 10 within a building, which uses the threat detection system to detect and thereby attempt to prevent threats to computing devices within its bounds. The first computer system 10 comprises three computers 1, 2, 3, a local server 4, and a multifunctional device (MFD) 5 that provides printing, scanning and facsimile functionalities to each of the computers 1, 2, 3. All of the devices within the first computer system 10 are communicatively coupled via a local area network (LAN) 6. Consequently, all the computers 1, 2, 3 can access the local server 4 via the LAN 6 and use the functionalities of the MFD 5 via the LAN 6.

**[0056]** The LAN 6 of the first computer system 10 is connected to the Internet 20, which in turn provides computers 1, 2, 3 with access to a multitude of other computing devices including server 30 and second computer system 40. Second computer system 40 also includes two computers 41, 42, connected by a second LAN 43.

**[0057]** In this exemplary embodiment of the invention, computer 1 on the first computer system 10 has the threat detection system and therefore runs the threat detection method for detecting threats to the first computer system. As such, it comprises a processor arranged to

run the steps of the process described herein, memory required to store information related to the running of the process, as well as a network interface for collecting the required information. This method shall now be described in detail with reference to Figure 3.

**[0058]** The computer 1 builds and maintains a dynamic, ever-changing model of the 'normal behavior' of each user and machine within the system 10. The approach is based on Bayesian mathematics, and monitors all interactions, events and communications within the system 10 - which computer is talking to which, files that have been created, networks that are being accessed.

**[0059]** For example, computer 2 is based in a company's San Francisco office and operated by a marketing employee who regularly accesses the marketing network. Computer 2 is active from about 8:30 AM until 6 PM and usually communicates with machines in the company's U.K. office in second computer system 40 between 9.30 AM and midday. The same employee virtually never accesses the employee time sheets, very rarely connects to the company's Atlanta network, and has no dealings in South-East Asia. The threat detection system takes all the information that is available relating to this employee to establish a 'pattern of life' for that person, which is dynamically updated as more information is gathered. The 'normal' model is used as a moving benchmark, allowing the system to spot behavior on a system that seems to fall outside of this normal pattern of life and to flag this behavior as anomalous, requiring further investigation.

**[0060]** The threat detection system is built to deal with the fact that today's attackers are getting stealthier. An attacker may be 'hiding' in a system to ensure that they avoid raising suspicion in an end user, such as by slowing their machine down, using normal software protocol. Any attack process thus stops or 'backs off automatically if the mouse or keyboard is used. However, yet more sophisticated attacks try the opposite, hiding in memory under the guise of a normal process and stealing CPU cycles only when the machine is active, to defeat a relatively-simple policing process. These sophisticated attackers look for activity that is not directly associated with the user's input. As an Advanced Persistent Threat (APT) attack typically has very long mission windows of weeks, months, or years, such processor cycles can be stolen so infrequently that they do not impact machine performance. However cloaked and sophisticated the attack is, the attack will always leave a measurable delta, even if extremely slight, in typical machine behavior, between pre and post compromise. This behavioral delta can be observed and acted on with the form of Bayesian mathematical analysis used by the threat detection system installed on the computer 1.

**[0061]** Figure 4 illustrates in a block diagram the integration of the threat detection system with other network protections. A network generally has a firewall 402 as a first line of defense. The firewall 402 analyzes packet headers on incoming network data packets to enforce network policy. The firewall 402 may be integrated with an intrusion prevention system (IPS) to analyze the packet header and payload for whole events. Internally, an identity management module 404 controls the access for the users of the network.

**[0062]** A network security module 406 can enforce practices and policies for the network as determined by a network administrator. An encryption module 408 can encrypt communications within the network, as well as encrypting and decrypting communications between network entities and outside entities. An anti-virus or anti-malware module 410 may search packets for known viruses and malware. A patch management module 412 can ensure that security applications within the network have applied the most up-to-date patches. A centralized logging module 414 may track communications both internal to and interactive with the network. The threat detection system can act as real time threat intelligence 416 for the network. The real time threat intelligence may interact with the other defense components to protect the network.

**[0063]** The cyber defense self-learning platform uses machine-learning technology. The machine-learning technology, using advanced mathematics, can detect previously unidentified threats, without rules, and automatically defend networks. Note, today's attacks can be of such severity and speed that a human response cannot happen quickly enough. Thanks to these self-learning advances, it is now possible for a machine to uncover emerging threats and deploy appropriate, real-time responses to fight back against the most serious cyber threats.

**[0064]** The cyber threat defense system builds a sophisticated 'pattern of life' - that understands what represents normality for every person, device, and network activity in the system being protected by the cyber threat defense system.

**[0065]** The threat detection system may self-learn and detect normality in order to spot true anomalies, allowing organizations of all sizes to understand the behavior of users and machines on their networks at both an individual and group level. Monitoring behaviors, rather than using predefined descriptive objects and/or signatures, means that more attacks can be spotted ahead of time and extremely subtle indicators of wrongdoing can be detected. Unlike traditional legacy defenses, a specific attack type or new malware does not have to have been seen first before it can be detected. A behavioral defense approach mathematically models both machine and human activity behaviorally, at and after the point of compromise, in order to predict and catch today's increasingly sophisticated cyber-attack vectors. The approach may thus computationally establish what is normal, in order to then detect what is abnormal.

**[0066]** This intelligent system may make value judgments and carry out higher value, more thoughtful tasks. Machine-learning requires complex algorithms to be devised and an overarching framework to interpret the results produced. However, when applied correctly these approaches can facilitate machines to make logical, probability-based decisions and undertake thoughtful tasks.

**[0067]** Advanced machine-learning is at the forefront of the fight against automated and human-driven cyber-threats, overcoming the limitations of rules and signature-based approaches. For example, the machine-learning learns what is normal within a network without depending upon knowledge of previous attacks. The machine-learning thrives on the scale, complexity, and diversity of modern businesses, where every device and person is slightly

different. The machine-learning turns the innovation of attackers against them, so that any unusual activity is visible. The machine-learning constantly revisits assumptions about behavior, using probabilistic mathematics. The machine-learning is always up to date and not reliant on human input. Utilizing machine-learning in cyber security technology is difficult, but when correctly implemented it is extremely powerful. The machine-learning means that previously unidentified threats can be detected, even when their manifestations fail to trigger any rule set or signature. Instead, machine-learning allows the system to analyze large sets of data and learn a 'pattern of life' for what it sees.

**[0068]** Figure 5 illustrates an application of a cyber threat defense system using advanced machine-learning to detect anomalous behavior. A normal pattern of behavior 510 may describe a set of user or device behavior within a threshold level of occurrence, such as a 98% probability of occurrence based on prior behavior. An anomalous activity 520 may describe a set of user or device behavior that is above the threshold level of occurrence. The cyber threat defense system can initiate an autonomous response 530 to counteract the anomalous activity, leaving the normal behavior unaffected.

**[0069]** Machine-learning can approximate some human capabilities to machines. Machine-learning can approximate thought by using past information and insights to form judgments. Machine-learning can act in real time so that the system processes information as it goes. Machine-learning can self-improve by constantly challenging and adapting the model's machine-learning understanding based on new information.

**[0070]** New unsupervised machine-learning therefore allows computers to recognize evolving threats, without prior warning or supervision.

### **Unsupervised machine-learning**

**[0071]** Unsupervised learning works things out without pre-defined labels. This allows the system to handle the unexpected and embrace uncertainty. The system does not always know the characteristics of the target of the search but can independently classify data and detect compelling patterns.

**[0072]** The cyber threat defense system's unsupervised machine-learning methods do not require training data with pre-defined labels. Instead, unsupervised machine-learning methods can identify key patterns and trends in the data, without the need for human input. Unsupervised learning provides the advantage of allowing computers to go beyond what their programmers already know and discover previously unknown relationships.

**[0073]** The cyber threat defense system uses unique implementations of unsupervised machine-learning algorithms to analyze network data at scale, intelligently handle the unexpected, and embrace uncertainty. Instead of relying on knowledge of past threats to be able to know what to look for, the cyber threat defense system may independently classify data

and detect compelling patterns that define what may be considered to be normal behavior. Any new behaviors that deviate from this notion of 'normality' may indicate threat or compromise. The impact of the cyber threat defense system's unsupervised machine-learning on cyber security is transformative. Threats from within, which would otherwise go undetected, can be spotted, highlighted, contextually prioritized, and isolated using these algorithms. The application of machine-learning has the potential to provide total network visibility and far greater detection levels, ensuring that networks have an internal defense mechanism. Machine-learning has the capability to learn when to execute automatic responses against the most serious cyber threats, disrupting in progress attacks before they become a crisis for the organization.

**[0074]** This new mathematics not only identifies meaningful relationships within data, but also quantifies the uncertainty associated with such inference. By knowing and understanding this uncertainty, it becomes possible to bring together many results within a consistent framework - the basis of Bayesian probabilistic analysis. The mathematics behind machine-learning is extremely complex and difficult to get right. Robust, dependable algorithms are developed, with a scalability that enables their successful application to real-world environments.

### **Overview**

**[0075]** In an embodiment, the cyber threat defense system's probabilistic approach to cyber security is based on a Bayesian framework. This allows the cyber threat defense system to integrate a huge number of weak indicators of potentially anomalous network behavior to produce a single clear measure of how likely a network device is to be compromised. This probabilistic mathematical approach provides an ability to understand important information amid the noise of the network, even when the target of a search is unknown.

### **Ranking threats**

**[0076]** Crucially, the cyber threat defense system's approach accounts for the inevitable ambiguities that exist in data, distinguishing between the subtly differing levels of evidence that different pieces of data may contain. Instead of generating the simple binary outputs 'malicious' or 'benign,' the cyber threat defense system's mathematical algorithms produce outputs that indicate differing degrees of potential compromise. This output enables users of the system to rank different alerts in a rigorous manner, prioritizing those that most urgently require action and simultaneously removing the problem of numerous false positives associated with a rule-based approach.

**[0077]** On a core level, the cyber threat defense system mathematically characterizes what constitutes 'normal' behavior based on the analysis of a large number of different measures of network behavior by a device. Such network behavior may include server access, data access,

timings of events, credential use, domain name server (DNS) requests, and other similar parameters. Each measure of network behavior is then monitored in real time to detect anomalous behaviors.

### **Clustering**

**[0078]** To be able to properly model what should be considered as normal for a device, the behavior of the device must be analyzed in the context of other similar devices on the network. To accomplish this, the cyber threat defense system leverages the power of unsupervised learning to algorithmically identify naturally occurring groupings of devices, a task which is impossible to do manually on even modestly sized networks.

**[0079]** In order to achieve as holistic a view of the relationships within the network as possible, the cyber threat defense system simultaneously employs a number of different clustering methods including matrix-based clustering, density based clustering, and hierarchical clustering techniques. The resulting clusters are then used to inform the modeling of the normative behaviors of individual devices. Clustering analyzes behavior in the context of other similar devices on the network. Clustering algorithms identify naturally occurring groupings of devices, which is impossible to do manually. Further, the cyber threat defense system may simultaneously run multiple different clustering methods to inform the models.

**[0080]** The inoculation module may also use clustering to identify which devices to send an inoculation notice. The inoculation module may select devices that have similar characteristics relevant to the anomalous event.

### **Network topology**

**[0081]** Any cyber threat detection system must also recognize that a network is far more than the sum of its individual parts, with much of its meaning contained in the relationships among its different entities. Plus, any cyber threat defense system must further recognize that complex threats can often induce subtle changes in this network structure. To capture such threats, the cyber threat defense system employs several different mathematical methods in order to be able to model multiple facets of a network topology.

**[0082]** One approach is based on iterative matrix methods that reveal important connectivity structures within the network. In tandem with these, the cyber threat defense system has developed innovative applications of models from the field of statistical physics, which allow the modeling of a network's 'energy landscape' to reveal anomalous substructures that may be concealed within.

### **Network structure**

**[0083]** A further important challenge in modeling the behaviors of network devices, as well as of networks themselves, is the high-dimensional structure of the problem with the existence of a huge number of potential predictor variables. Observing packet traffic and host activity within an enterprise local area network (LAN), wide area network (WAN) and Cloud is difficult because both input and output can contain many inter-related features, such as protocols, source and destination machines, log changes, rule triggers, and others. Learning a sparse and consistent structured predictive function is crucial to avoid over fitting.

**[0084]** In this context, the cyber threat defense system has employed a cutting edge large-scale computational approach to learn sparse structure in models of network behavior and connectivity based on applying L1-regularization techniques, such as a Least Absolute Shrinkage and Selection Operator (LASSO) method. This allows for the discovery of true associations between different network components and events that can be cast as efficiently solvable convex optimization problems and yield parsimonious models.

### **Recursive Bayesian Estimation**

**[0085]** To combine these multiple analyses of different measures of network behavior to generate a single comprehensive picture of the state of each device, the cyber threat defense system takes advantage of the power of Recursive Bayesian Estimation (RBE) via an implementation of the Bayes filter.

**[0086]** Using RBE, the cyber threat defense system's mathematical models can constantly adapt themselves, in a computationally efficient manner, as new information becomes available to the system. They continually recalculate threat levels in the light of new evidence, identifying changing attack behaviors where conventional signature-based methods fail.

**[0087]** The cyber threat defense system's innovative approach to cyber security has pioneered the use of Bayesian methods for tracking changing device behaviors and computer network structures. The core of the cyber threat defense system's mathematical modeling is the determination of normative behavior, enabled by a sophisticated software platform that allows for its mathematical models to be applied to new network data in real time. The result is a system that can identify subtle variations in machine events within a computer networks behavioral history that may indicate cyber-threat or compromise.

**[0088]** The cyber threat defense system uses mathematical analysis and machine-learning to detect potential threats, allowing the system to stay ahead of evolving risks. The cyber threat defense system approach means that detection no longer depends on an archive of previous attacks. Instead, attacks can be spotted against the background understanding of what represents normality within a network. No pre-definitions are needed, which allows for the best possible insight and defense against today's threats. On top of the detection capability, the

cyber threat defense system can create digital antibodies automatically, as an immediate response to the most threatening cyber breaches. The cyber threat defense system approach both detects and defends against cyber threat. Genuine unsupervised machine-learning eliminates the dependence on signature-based approaches to cyber security, which are not working. The cyber threat defense system's technology can become a vital tool for security teams attempting to understand the scale of their network, observe levels of activity, and detect areas of potential weakness. These no longer need to be manually sought out, but rather are flagged by the automated system and ranked in terms of their significance.

**[0089]** Machine-learning technology is the fundamental ally in the defense of systems from the hackers and insider threats of today, and in formulating response to unknown methods of cyber-attack. It is a momentous step change in cyber security. Defense must start within.

### **An example method**

**[0090]** The threat detection system shall now be described in further detail with reference to a flow of the process carried out by the threat detection system for automatic detection of cyber threats through probabilistic change in normal behavior through the application of an unsupervised Bayesian mathematical model to detect behavioral change in computers and computer networks.

**[0091]** The core threat detection system is termed the 'Bayesian probabilistic'. The Bayesian probabilistic is a Bayesian system of automatically determining periodicity in multiple time series data and identifying changes across single and multiple time series data for the purpose of anomalous behavior detection.

**[0092]** Figure 6 illustrates a flowchart of an embodiment of a method for modeling human, machine or other activity. The cyber threat defense system initially ingests data from multiple sources (Block 602). The raw data sources include, but are not limited to raw network Internet Protocol (IP) traffic captured from an IP or other network Test Access Points (TAP) or Switched Port Analyzer (SPAN) port; machine generated log files; building access ("swipe card") systems; IP or non-IP data flowing over an Industrial Control System (ICS) distributed network; individual machine, peripheral or component power usage; telecommunication signal strength; or machine level performance data taken from on-host sources, such as central processing unit (CPU) usage, memory usage, disk usage, disk free space, network usage, and others.

**[0093]** The cyber threat defense system derives second order metrics from that raw data (Block 604). From these raw sources of data, multiple metrics can be derived, each producing time series data for the given metric. The data are bucketed into individual time slices. For example, the number observed could be counted per 1 second, per 10 seconds or per 60 seconds. These buckets can be combined at a later stage where required to provide longer range values for any multiple of the chosen internal size. For example, if the underlying time slice chosen is 60 seconds long, and thus each metric time series stores a single value for the

metric every 60 seconds, then any new time series data of a fixed multiple of 60 seconds (such as 120 seconds, 180 seconds, 600 seconds etc.) can be computed with no loss of accuracy. Metrics are chosen directly and fed to the Bayesian probabilistic by a lower order model which reflects some unique underlying part of the data, and which can be derived from the raw data with particular domain knowledge. The metrics that are obtained depends on the threats that the system is looking for. In order to provide a secure system, the cyber threat defense system commonly obtains multiple metrics relating to a wide range of potential threats. Communications from components in the network contacting known suspect domains.

**[0094]** The actual specific metrics used are largely irrelevant to the Bayesian probabilistic system, as long as a metric is selected. Metrics derived from network traffic could include data such as the number of bytes of data entering or leaving a networked device per time interval, file access, the commonality or rarity of a communications process, an invalid secure-sockets layer (SSL) certification, a failed authorization attempt, or email access patterns.

**[0095]** In the case where transmission control protocol (TCP), user datagram protocol (UDP), or other Transport Layer IP protocols are used over the IP network, and in cases where alternative Internet Layer protocols are used, such as Internet Control Message Protocol (ICMP) or Internet Group Message Protocol (IGMP), knowledge of the structure of the protocol in use and basic packet header analysis can be utilized to generate further metrics. Such further metrics may include the number of multicasts per time interval originating from a networked device and intended to reach publicly addressable IP ranges, the number of internal link-local IP Broadcast requests originating from a networked device, the size of the packet payload data, or the number of individual TCP connections made by a device, or data transferred by a device, either as a combined total across all destinations or to any definable target network range, such as a single target machine or a specific network range.

**[0096]** In the case of IP traffic where the Application Layer protocol can be determined and analyzed, further types of time series metric can be defined. These time series metrics may include, for example, the number of DNS requests a networked device generates per time interval, again either to any definable target network range or in total; the number of Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP) or Internet Message Access Protocol (IMAP) logins or login failures a machine generates per time interval; the number of Lightweight Directory Access Protocol (LDAP) logins or login failures generated; data transferred via file sharing protocols such as Server Message Block (SMB), SMB2, File Transfer Protocol (FTP), or others; or logins to Microsoft Windows Active Directory, Secure Shell (SSH) or Local Logins to Linux or Unix-like systems, or other authenticated systems such as Kerberos.

**[0097]** The raw data required to obtain these metrics may be collected via a passive fiber or copper connection to the networks internal switch gear, from virtual switching implementations, cloud-based systems, or communicating devices themselves. Ideally, the system receives a copy of every communications packet to provide full coverage of an organization.

**[0098]** For other sources, a number of domain specific time series data are derived, each chosen to reflect a distinct and identifiable facet of the underlying source of the data, which in some way reflects the usage or behavior of that system over time.

**[0099]** Many of these time series data sets are extremely sparse, with most data points equal to 0. Examples would be employee's using swipe cards to access a building or part of a building, or user's logging into their workstation, authenticated by Microsoft Windows Active Directory Server, which is typically performed a small number of times per day. Other time series data sets are much more populated, such as, the size of data moving to or from an always-on Web Server, the Web Servers CPU utilization, or the power usage of a photocopier.

**[0100]** Regardless of the type of data, such time series data sets, whether originally produced as the result of explicit human behavior or an automated computer or other system to exhibit periodicity, have the tendency for various patterns within the data to recur at approximately regular intervals. Furthermore, such data may have many distinct but independent regular time periods apparent within the time series.

**[0101]** Detectors carry out analysis of the second order metrics (Block 606). Detectors are discrete mathematical models that implement a specific mathematical method against different sets of variables with the target network. For example, Hidden Markov Models (HMM) may look specifically at the size and transmission time of packets between nodes. The detectors are provided in a hierarchy that is a loosely arranged pyramid of models. Each detector model effectively acts as a filter and passes its output to another model higher up the pyramid. At the top of the pyramid is the Bayesian probabilistic that is the ultimate threat decision making model. Lower order detectors each monitor different global attributes or 'features' of the underlying network and or computers. These attributes may be value over time for all internal computational features such as packet velocity and morphology, endpoint file system values, and TCP/IP protocol timing and events. Each detector is specialized to record and make decisions on different environmental factors based on the detectors own internal mathematical model such as an HMM.

**[0102]** While the threat detection system may be arranged to look for any possible threat, in practice the system may keep watch for one or more specific threats depending on the network in which the threat detection system is being used. For example, the threat detection system provides a way for known features of the network such as desired compliance and Human Resource policies to be encapsulated in explicitly defined heuristics or detectors that can trigger when in concert with set or moving thresholds of probability abnormality coming from the probability determination output. The heuristics are constructed using complex chains of weighted logical expressions manifested as regular expressions with atomic objects that are derived at run time from the output of data measuring/tokenizing detectors and local contextual information. These chains of logical expression are then stored in online libraries and parsed in real-time against output from the measures/tokenizing detectors. An example policy could take the form of "alert me if any employee subject to HR disciplinary circumstances (contextual information) is accessing sensitive information (heuristic definition) in a manner that is

anomalous when compared to previous behavior (Bayesian probabilistic output)". In other words, different arrays of pyramids of detectors are provided for detecting particular types of threats.

**[0103]** The analysis performed by the detectors on the second order metrics then outputs data in a form suitable for use with the model of normal behavior. As will be seen, the data is in a form suitable for comparing with the model of normal behavior and for updating the model of normal behavior.

**[0104]** The threat detection system computes a threat risk parameter indicative of a likelihood of there being a threat using automated adaptive periodicity detection mapped onto observed behavioral pattern-of-life analysis (Block 608). This deduces that a threat over time exists from a collected set of attributes that themselves have shown deviation from normative collective or individual behavior. The automated adaptive periodicity detection uses the period of time the Bayesian probabilistic has computed to be most relevant within the observed network or machines. Furthermore, the pattern of life analysis identifies how a human or machine behaves over time, such as when they typically start and stop work. Since these models are continually adapting themselves automatically, they are inherently harder to defeat than known systems. The threat risk parameter is a probability of there being a threat in certain arrangements. Alternatively, the threat risk parameter is a value representative of there being a threat, which is compared against one or more thresholds indicative of the likelihood of a threat.

**[0105]** In practice, the step of computing the threat involves comparing current data collected in relation to the user with the model of normal behavior of the user and system being analyzed. The current data collected relates to a period in time, this could be in relation to a certain influx of new data or a specified period of time from a number of seconds to a number of days. In some arrangements, the system is arranged to predict the expected behavior of the system. The expected behavior is then compared with actual behavior in order to determine whether there is a threat.

**[0106]** The system uses machine-learning or Artificial Intelligence to understand what is normal inside a company's network, and when something's not normal. The system then invokes automatic responses to disrupt the cyber-attack until the human team can catch up. This could include interrupting connections, preventing the sending of malicious emails, preventing file access, preventing communications outside of the organization, etc. The approach begins in as surgical and directed way as possible to interrupt the attack without affecting the normal behavior of, for example, a laptop. If the attack escalates, the cyber threat defense system may ultimately quarantine a device to prevent wider harm to an organization.

**[0107]** In order to improve the accuracy of the system, a check can be carried out in order to compare current behavior of a user with associated users, such as users within a single office. For example, if there is an unexpectedly low level of activity from a user, this may not be due to unusual activity from the user, but rather a factor affecting the office as a whole. Various other

factors can be considered in order to assess whether abnormal behavior is actually indicative of a threat.

**[0108]** Finally, the cyber threat defense system determines, based on the threat risk parameter, as to whether further action need be taken regarding the threat (Block 610). A human operator may make this determination after being presented with a probability of there being a threat. Alternately, an algorithm may make the determination, such as by comparing the determined probability with a threshold.

**[0109]** In one arrangement, given the unique global input of the Bayesian probabilistic, a form of threat visualization is provided in which the user can view the threat landscape across all internal traffic and do so without needing to know how their internal network is structured or populated and in such a way as a 'universal' representation is presented in a single pane no matter how large the network. A topology of the network under scrutiny is projected automatically as a graph based on device communication relationships via an interactive 3D user interface. The projection can scale linearly to any node scale without prior seeding or skeletal definition.

**[0110]** The threat detection system that has been discussed above therefore implements a propriety form of recursive Bayesian estimation to maintain a distribution over the probability state variable. This distribution is built from the complex set of low-level host, network, and traffic observations or 'features'. These features are recorded iteratively and processed in real time on the platform. A plausible representation of the relational information among entities in dynamic systems in general, such as an enterprise network, a living cell or a social community, or indeed the entire internet, is a stochastic network, which is topological rewiring and semantically evolving over time. In many high-dimensional structured input/output problems, such as the observation of packet traffic and host activity within a distributed digital enterprise, where both input and output can contain tens of thousands to millions of interrelated features (data transport, host-web-client dialogue, log change and rule trigger, etc.), learning a sparse and consistent structured predictive function is challenged by a lack of normal distribution. To overcome this, the threat detection system comprise a data structure that decides on a rolling continuum rather than a stepwise method in which recurring time cycles, such as the working day, shift patterns, and other routines are dynamically assigned, thus providing a non-frequentist architecture for inferring and testing causal links between explanatory variables, observations and feature sets. This permits an efficiently solvable convex optimization problem and yield parsimonious models. In such an arrangement, the threat detection processing may be triggered by the input of new data. Alternatively, the threat detection processing may be triggered by the absence of expected data. In some arrangements, the processing may be triggered by the presence of a particular actionable event.

**[0111]** The method and system are arranged to be performed by one or more processing components with any portions of software stored in an executable format on a computer readable medium. The computer readable medium may be non-transitory and does not include radio or other carrier waves. The computer readable medium could be, for example, a

physical computer readable medium such as semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disc, and an optical disk, such as a CD-ROM, CD-R/W or DVD.

**[0112]** The various methods described above may be implemented by a computer program product. The computer program product may include computer code arranged to instruct a computer to perform the functions of one or more of the various methods described above. The computer program and/or the code for performing such methods may be provided to an apparatus, such as a computer, on a computer readable medium or computer program product. For the computer program product, a transitory computer readable medium may include radio or other carrier waves.

**[0113]** An apparatus such as a computer may be configured in accordance with such code to perform one or more processes in accordance with the various methods discussed herein.

### **Inoculations**

**[0114]** The cyber threat defense system can leverage the identified anomalous events to strengthen defenses within the network and at affiliated networks. Figure 7 illustrates an inoculation component of the cyber threat defense system. An analyst using the cyber threat defense system can discover signs of an in-progress attack in a digital network of an organization (Block 702). The analyst may be a human expert or may be an analyst program that uses machine-learning to identify an anomalous event, resulting in a totally automated end-to-end attack detection, inoculation distribution, and autonomous defense implementation.

**[0115]** The analyst may observe hallmarks of a digital attack in an anomalous event, referred to as an "Behavioral Indicators of Compromise" (b-IoCs). These hallmarks describe general patterns of anomalous behavior derived from network activity and constitute network artefacts such as, for example, a beaconing frequency score, a size of file that has a greater than and less than parameters, the length of time that has occurred between cyber threat alerts that are indicative of the malicious activity. Note, b-IoCs should be considered distinct from standard "Indicators of Compromise" which comprise inflexible, fixed values such as, for example, hashes, subdomains, domains, URLs and URIs. The analyst may use a researcher module operating an artificial intelligence (AI) algorithm to collect outside data about the anomalous event. The researcher module can assess whether the b-IoC has previously appeared in other published threat research or known lists of malicious files or internet addresses (Block 704). The analyst can review the outside data alongside the observed behavior of the attack. The analyst may then conclude that the attack is both novel and malicious (706).

**[0116]** The analyst documents their findings and adds explanatory notes regarding the digital hallmarks of the malicious action (Block 708). Additionally, the analyst adds in any recommendations of an appropriate response that an autonomous defense system may select in the event of such an attack. For example, the autonomous defense system can prevent an

infected device from speaking back to an attacker's control systems. Additionally, the autonomous defense system can interrupt any unusual file uploads going outside of the organization from the infected device.

**[0117]** The cyber threat defense system can create an inoculation package of the digital hallmarks, suggested autonomous responses, and documentation (Block 710). The cyber threat defense system can automatically disseminate this inoculation package to any other subscriber organization as an inoculation notice. Alternately, the cyber threat defense system can store the inoculation package as an inoculation record in a network- accessible inoculation database for future retrieval.

**[0118]** If the attack is already inside another organization or begins to try to get a foothold in an organization, that organization can compare its network behavioral information to those detailed in the inoculation package to identify shared behavioral indicators of compromise. The affiliated subscriber organization may generate an alert on that organization's security dashboard detailing the nature of the attack, along with supporting notes (Block 712). A human analyst from that organization may then investigate and respond.

**[0119]** If the organization has also deployed autonomous defenses, then that autonomous defense system can review the suggested response actions detailed in the inoculation package and select the most appropriate actions to take in the current situation. Typically, the autonomous defense system will start with a minimum level of disruption, aiming to surgically stop the attack without affecting normal organizational activity. If the attack changes in nature or becomes more aggressive, then the autonomous defense system can select more severe controls to apply to the infected device or devices. These actions could escalate up to removing internet access or even total network access.

**[0120]** In an embodiment, a mimetics system is configured to create and share digital antibodies for previously unknown cyber threats across different users' platforms. The mimetic system can share an inoculation package having digital antibodies with an autonomous response. The mimetics system and communication mechanism are configured to share a brand-new digital antibody against an unknown threat. When the mimetics system identifies an attack that no one has seen before, then the cyber threat defense system can use the intelligence models to determine what the appropriate autonomous response to this new unknown threat should be. Next, the mimetics system can encapsulate the identity of the threat, the characteristics to identify that threat, and the autonomous response to defend the network against this threat into a new digital antibody. The digital antibody can be shared or communicated and conveyed to the community at large or just to a selected segment of community, such as entities in that sector or subdivisions within the organization.

**[0121]** Figure 8 illustrates a flowchart of an embodiment of a method for developing an inoculation record based on the detection of an anomalous event. The cyber threat defense system can receive, via at least one input port, input data from a probe monitoring a network entity on a network (Block 802). The network entity can represent at least one of a user and a

device associated with a network. The cyber threat defense system has a cyber threat module configured to analyze the input data using at least one machine-learning model to spot behavior on the network deviating from a normal benign behavior of that network entity (Block 804). The at least one machine-learning model is trained on generic normal benign behavior of a generic network entity using a normal behavior benchmark describing parameters corresponding to a normal pattern of activity for that network entity. The cyber threat defense system has a researcher module configured to collect an outside data set describing at least one of an outside action and an outside state related to the input data from at least one data source outside the network (Block 806). The cyber threat defense system has a comparison module that compares the input data to the at least one machine-learning model to spot behavior on the network deviating from a normal benign behavior of that network entity (Block 808). The comparison module can identify whether the network entity is in a breach state of the normal behavior benchmark (Block 810). The cyber threat module can identify whether the breach state and a chain of relevant behavioral parameters deviating from the normal benign behavior of that network entity correspond to a cyber threat (Block 810). The cyber threat defense system has an inoculation module configured to generate an inoculation pattern describing the breach state and the chain of relevant behavioral parameters corresponding to the cyber threat identified by the cyber threat module (Block 814). The inoculation pattern abstracts the incident data of the cyber threat to more clearly identify the relevant features of the cyber threat. The inoculation module can store the inoculation pattern in an inoculation record in a network-accessible inoculation database in the data store (Block 816).

**[0122]** The cyber threat defense system can generate a threat risk parameter to describe the relative dangers of a cyber threat. Figure 9 illustrates a block diagram of a threat risk parameter. The threat risk parameter can have a confidence score 902 indicating a threat likelihood describing a probability that the breach state is the cyber threat. The threat risk parameter can have a severity score 904 indicating a percentage that the network entity in the breach state is deviating from normal behavior, as represented by the at least one model. The threat risk parameter can have a consequence score 906 indicating a severity of damage attributable to the cyber threat.

**[0123]** Figure 10 illustrates a flowchart of an embodiment of a method for generating a threat risk parameter. The cyber threat module can generate a threat risk parameter listing a set of values describing aspects of the breach state (Block 1002). The cyber threat module can generate a confidence score (Block 1004). The cyber threat module can generate a severity score (Block 1006). The cyber threat module can generate a consequence score (Block 1008). The cyber threat module can populate the threat risk parameter with at least one of the confidence score, the severity score, and the consequence score (Block 1010)

**[0124]** The cyber threat defense system can store information describing the identified cyber threats in an inoculation record. Figure 11 illustrates a block diagram of an inoculation record. The inoculation record can have a random number to act as a threat identifier 1102. The threat identifier 1102 may be disassociated from the network entity to protect personally identifiable information. The actual identity of the network entity can be obfuscated or anonymized to

remove any personally identifiable information for the network entity. The inoculation record has an inoculation pattern 1104 describing the breach state and the chain of relevant behavioral parameters corresponding to the cyber threat identified by the cyber threat module. The inoculation record has a threat risk parameter 1106 listing a set of values describing aspects of the breach state. The inoculation record has a context data set 1108 provided by the user analyst describing at least one of a network entity action and a network entity state related to the cyber threat to provide context for the actions and states of the network entity during the breach state. The inoculation record has an outside data set 1110 an outside data set collected from at least one data source outside the network describing at least one of an outside action and an outside state related to the cyber threat. The inoculation record has a remediation action instruction 1112 to describe at least one action to remediate the cyber threat.

**[0125]** Figure 12 illustrates a flowchart of an embodiment of a method for generating an inoculation record. The cyber threat module can generate an inoculation pattern describing the breach state and the chain of relevant behavioral parameters corresponding to the cyber threat identified by the cyber threat module (Block 1202). The inoculation record can associate the inoculation pattern with a random number acting as a threat identifier (Block 1204). The inoculation module can anonymize the incident data in creating an inoculation pattern to remove any personally identifiable information for the network entity from the inoculation pattern (Block 1206). The inoculation record associates the inoculation pattern with a context data set describing at least one of a network entity action and a network entity state related to the cyber threat provided by a user analyst (Block 1208). The inoculation record associates the inoculation pattern with an outside data set collected from at least one data source outside the network describing at least one of an outside action and an outside state related to the cyber threat (Block 1210). The inoculation module can generate a remediation action instruction to describe at least one action to remediate the cyber threat (Block 1212). The inoculation record associates the inoculation pattern with the remediation action instruction (Block 1214). The inoculation module is configured to update the inoculation pattern in the inoculation record based on a subsequent event at the network entity or a target device (Block 1216).

**[0126]** The cyber threat defense system may allow the user analyst to send an inoculation notification. Figure 13 illustrates a flowchart of an embodiment of a method for sending an inoculation notice. The user interface module can present a description of the breach state to a user analyst for review (Block 1302). The user interface module can present the threat risk parameter listing a set of values describing aspects of the cyber threat to the user analyst (Block 1304). The inoculation module can compare the threat risk parameter to a benchmark vector describing various thresholds for normal behavior (Block 1306). The user interface module may receive, from the user analyst, a context data set describing at least one of a network entity action and a network entity state related to the cyber threat (Block 1308). The user interface module can receive at least one of a triggering input and a blocking input from a user analyst (Block 1310). The triggering input directs transmission of the inoculation notice to the target device. The blocking input prevents transmission of the inoculation notice to the target device.

**[0127]** The inoculation module is configured to create an entity cluster to group the network entity with other entities of the network based on the chain of relevant behavior parameters of the inoculation pattern (Block 1312). To be able to properly identify a target device that may benefit from inoculation, a cyber threat defense system may identify other similar devices on the network. To accomplish this, the cyber threat defense system can leverage the power of unsupervised learning to algorithmically identify naturally occurring groupings of devices, a task which is impossible to do manually on even modestly sized networks.

**[0128]** In order to achieve as holistic a view of the relationships within the network as possible, the cyber threat defense system simultaneously employs a number of different clustering methods, such as matrix-based clustering, density-based clustering, and hierarchical clustering techniques. An inoculation module may use the resulting clusters to identify individual devices that may benefit from the clustering. The inoculation module is configured to select a target device for notification regarding the inoculation pattern based on the entity cluster (Block 1314).

**[0129]** The inoculation module is configured to generate an inoculation notice having the inoculation pattern describing the breach state and the chain of relevant behavioral parameters to warn of a potential cyber threat (Block 1316). The inoculation module is configured to use the communication module to send the inoculation notice to a target device to warn of a potential cyber threat by the target device (Block 1318). The inoculation module is configured to send a remediation action instruction in the inoculation notice to describe at least one action to remediate the cyber threat (Block 1320).

**[0130]** The cyber threat defense system can use a benchmark to determine whether a user or device is in a breach state. The cyber threat defense system can also use a similar benchmark to automatically determine whether to send an inoculation notice without user analyst input. The cyber threat defense system can present the benchmark to the user analyst to provide context for the threat risk parameter. Alternately, the cyber threat defense system can use the benchmark to automatically determine whether to send an inoculation notice. The inoculation module may use a benchmark matrix representing a varied set of benchmarks to consider a variety of factors when determining whether to send an inoculation notice.

**[0131]** Figure 14 illustrates a block diagram of a benchmark matrix. The inoculation module, in conjunction with the cyber threat module, can populate the benchmark matrix with moving benchmarks that can adapt to the changing nature of both the network and threats to the network. The benchmark matrix can have a confidence benchmark 1402 indicating a threat likelihood describing a probability that the breach state is the cyber threat. The benchmark matrix can have a severity benchmark 1404 indicating a percentage above which the network entity is in the breach state. The benchmark matrix can have a consequence benchmark 1406 indicating a severity of damage attributable to the cyber threat that above which immediate action is to be taken. The inoculation module can adjust these benchmarks as more data is added and greater user input is received.

**[0132]** The inoculation module can assign a weight to each bench mark score to assign a relative importance to each bench mark score to factor in a decision to send an inoculation notice. As with the benchmarks, these weights may evolve over time. For example, the benchmark matrix can have a confidence weight 1408 indicating the importance of the confidence benchmark, a severity weight 1410 indicating the importance of the severity benchmark, and a consequence weight 1412 indicating the importance of the consequence benchmark. Using these assigned weights, different deviations from the benchmarks may have a greater result on the final decision to send and inoculation notice.

**[0133]** Figure 15 illustrates a flowchart of an embodiment of a method for comparing analyzed input data to a benchmark to trigger an inoculation notice. The inoculation module can generate a benchmark matrix having a set of benchmark scores to determine whether to send the inoculation notice (Block 1502). The inoculation module can populate the benchmark scores in the benchmark matrix based on data gathered during the breach identification and inoculation process (Block 1504). The inoculation module can assign a weight to each bench mark score to assign a relative importance to each benchmark score (Block 1506). The cyber threat module can generate a threat risk parameter listing a set of values describing aspects of the cyber threat (Block 1508). The cyber threat module can populate the threat risk parameter with at least one of a confidence score, a severity score, and a consequence score (Block 1510). The inoculation module can compare the threat risk parameter to the benchmark matrix to determine whether to send the inoculation notice (Block 1512).

**[0134]** While generally the inoculation module sends the inoculation notice immediately, the inoculation module may store an inoculation record describing anomalous events for later retrieval by a target device. Figure 16 illustrates a flowchart of an embodiment of a method for sending an inoculation report. The inoculation module is configured to generate an inoculation report listing at least the inoculation record (Block 1602). The inoculation module is configured to create an entity cluster to group the network entity with other entities of the network based on the chain of relevant behavior parameters of the inoculation pattern (Block 1604). The inoculation module is configured to select a target device to receive an inoculation report regarding the inoculation record based on the entity cluster (Block 1606). The inoculation module is configured to send the inoculation report to a target device related to the network entity to instruct the target device to retrieve the inoculation record from the network-accessible inoculation database (Block 1608). The inoculation module can receive an inoculation request for the inoculation record form the target device (Block 1610). The inoculation module may then provide the inoculation record to the target device (Block 1612).

### **Web Site**

**[0135]** The web site is configured as a browser-based tool or direct cooperating app tool for configuring, analyzing, and communicating with the cyber threat defense system.

### **Network**

**[0136]** A number of electronic systems and devices can communicate with each other in a network environment. Figure 17 illustrates in a simplified diagram a networked environment. The network environment has a communications network. The network can include one or more networks selected from an optical network, a cellular network, the Internet, a Local Area Network ("LAN"), a Wide Area Network ("WAN"), a satellite network, a 3<sup>rd</sup> party 'cloud' environment; a fiber network, a cable network, and combinations thereof. In some embodiments, the communications network is the Internet. There may be many server computing systems and many client computing systems connected to each other via the communications network.

**[0137]** The communications network can connect one or more server computing systems selected from at least a first server computing system and a second server computing system to each other and to at least one or more client computing systems as well. The server computing systems can each optionally include organized data structures such as databases. Each of the one or more server computing systems can have one or more virtual server computing systems, and multiple virtual server computing systems can be implemented by design. Each of the one or more server computing systems can have one or more firewalls and similar defenses to protect data integrity.

**[0138]** At least one or more client computing systems for example, a mobile computing device (e.g., smartphone with an Android-based operating system) can communicate with the server(s). The client computing system can include, for example, the software application or the hardware-based system in which may be able to exchange communications with the first electric personal transport vehicle, and/or the second electric personal transport vehicle. Each of the one or more client computing systems can have one or more firewalls and similar defenses to protect data integrity.

**[0139]** A cloud provider platform may include one or more of the server computing systems. A cloud provider can install and operate application software in a cloud (e.g., the network such as the Internet) and cloud users can access the application software from one or more of the client computing systems. Generally, cloud users that have a cloud-based site in the cloud cannot solely manage a cloud infrastructure or platform where the application software runs. Thus, the server computing systems and organized data structures thereof can be shared resources, where each cloud user is given a certain amount of dedicated use of the shared resources. Each cloud user's cloud-based site can be given a virtual amount of dedicated space and bandwidth in the cloud. Cloud applications can be different from other applications in their scalability, which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand. Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user, who sees only a single access point.

**[0140]** Cloud-based remote access can be coded to utilize a protocol, such as Hypertext Transfer Protocol ("HTTP"), to engage in a request and response cycle with an application on a client computing system such as a web-browser application resident on the client computing system. The cloud-based remote access can be accessed by a smartphone, a desktop computer, a tablet, or any other client computing systems, anytime and/or anywhere. The cloud-based remote access is coded to engage in 1) the request and response cycle from all web browser based applications, 3) the request and response cycle from a dedicated on-line server, 4) the request and response cycle directly between a native application resident on a client device and the cloud-based remote access to another client computing system, and 5) combinations of these.

**[0141]** In an embodiment, the server computing system can include a server engine, a web page management component, a content management component, and a database management component. The server engine can perform basic processing and operating-system level tasks. The web page management component can handle creation and display or routing of web pages or screens associated with receiving and providing digital content and digital advertisements. Users (e.g., cloud users) can access one or more of the server computing systems by means of a Uniform Resource Locator ("URL") associated therewith. The content management component can handle most of the functions in the embodiments described herein. The database management component can include storage and retrieval tasks with respect to the database, queries to the database, and storage of data.

**[0142]** In some embodiments, a server computing system can be configured to display information in a window, a web page, or the like. An application including any program modules, applications, services, processes, and other similar software executable when executed on, for example, the server computing system, can cause the server computing system to display windows and user interface screens in a portion of a display screen space. With respect to a web page, for example, a user via a browser on the client computing system can interact with the web page, and then supply input to the query/fields and/or service presented by the user interface screens. The web page can be served by a web server, for example, the server computing system, on any Hypertext Markup Language ("HTML") or Wireless Access Protocol ("WAP") enabled client computing system (e.g., the client computing system 802B) or any equivalent thereof. The client computing system can host a browser and/or a specific application to interact with the server computing system. Each application has a code scripted to perform the functions that the software component is coded to carry out such as presenting fields to take details of desired information. Algorithms, routines, and engines within, for example, the server computing system can take the information from the presenting fields and put that information into an appropriate storage medium such as a database (e.g., database). A comparison wizard can be scripted to refer to a database and make use of such data. The applications may be hosted on, for example, the server computing system and served to the specific application or browser of, for example, the client computing system. The applications then serve windows or pages that allow entry of details.

### **Computing systems**

**[0143]** A computing system can be, wholly or partially, part of one or more of the server or client computing devices in accordance with some embodiments. Components of the computing system can include, but are not limited to, a processing unit having one or more processing cores, a system memory, and a system bus that couples various system components including the system memory to the processing unit. The system bus may be any of several types of bus structures selected from a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures.

**[0144]** The computing system typically includes a variety of computing machine-readable media. Computing machine-readable media can be any available media that can be accessed by computing system and includes both volatile and nonvolatile media, and removable and non-removable media. By way of example, and not limitation, computing machine-readable media use includes storage of information, such as computer-readable instructions, data structures, other executable software or other data. Computer-storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other tangible medium which can be used to store the desired information and which can be accessed by the computing device. Transitory media, such as wireless channels, are not included in the machine-readable media. Communication media typically embody computer readable instructions, data structures, other executable software, or other transport mechanism and includes any information delivery media.

**[0145]** The system memory includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) and random access memory (RAM). A basic input/output system (BIOS) containing the basic routines that help to transfer information between elements within the computing system, such as during start-up, is typically stored in ROM. RAM typically contains data and/or software that are immediately accessible to and/or presently being operated on by the processing unit. By way of example, and not limitation, the RAM can include a portion of the operating system, application programs, other executable software, and program data.

**[0146]** The drives and their associated computer storage media discussed above, provide storage of computer readable instructions, data structures, other executable software and other data for the computing system.

**[0147]** A user may enter commands and information into the computing system through input devices such as a keyboard, touchscreen, or software or hardware input buttons, a microphone, a pointing device and/or scrolling input component, such as a mouse, trackball or touch pad. The microphone can cooperate with speech recognition software. These and other input devices are often connected to the processing unit through a user input interface that is coupled to the system bus, but can be connected by other interface and bus structures, such

as a parallel port, game port, or a universal serial bus (USB). A display monitor or other type of display screen device is also connected to the system bus via an interface, such as a display interface. In addition to the monitor, computing devices may also include other peripheral output devices such as speakers, a vibrator, lights, and other output devices, which may be connected through an output peripheral interface.

**[0148]** The computing system can operate in a networked environment using logical connections to one or more remote computers/client devices, such as a remote computing system. The logical connections can include a personal area network ("PAN") (e.g., Bluetooth<sup>®</sup>), a local area network ("LAN") (e.g., Wi-Fi), and a wide area network ("WAN") (e.g., cellular network), but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet. A browser application or direct app corresponding with a cloud platform may be resident on the computing device and stored in the memory.

**[0149]** It should be noted that the present design can be carried out on a single computing system and/or on a distributed system in which different portions of the present design are carried out on different parts of the distributed computing system.

**[0150]** Note, an application described herein includes but is not limited to software applications, mobile apps, and programs that are part of an operating system application. Some portions of this description are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. These algorithms can be written in a number of different software programming languages such as Python, C, C+, or other similar languages. Also, an algorithm can be implemented with lines of code in software, configured logic gates in software, or a combination of both. In an embodiment, the logic consists of electronic circuits that follow the rules of Boolean Logic, software that contain patterns of instructions, or any combination of both.

**[0151]** It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the above discussions, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the

computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers, or other such information storage, transmission or display devices.

**[0152]** Many functions performed by electronic hardware components can be duplicated by software emulation. Thus, a software program written to accomplish those same functions can emulate the functionality of the hardware components in input-output circuitry. Also, multiple modules can be combined into a single module where logically possible.

**[0153]** While the foregoing design and embodiments thereof have been provided in considerable detail, it is not the intention of the applicant(s) for the design and embodiments provided herein to be limiting. Additional adaptations and/or modifications are possible, and, in broader aspects, these adaptations and/or modifications are also encompassed. Accordingly, departures may be made from the foregoing design and embodiments without departing from the scope afforded by the following claims, which scope is only limited by the claims when appropriately construed.

## REFERENCES CITED IN THE DESCRIPTION

Cited references

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

**Patent documents cited in the description**

- [US20170118236A](#) [0004]
- [US20170262633A](#) [0005]
- [US20160261621A](#) [0006]

**P a t e n t k r a v**

1. Fremgangsmåde til et cybertrusselforsvarssystem (100), omfattende:

5 sammenligning af inputdata, der overvåger en netværksenhed, med mindst én maskinlæringsmodel, der er trænet i en normal godartet adfærd hos netværksenheden ved hjælp af et benchmark for normal adfærd, der beskriver parametre, der svarer til et normalt aktivitetsmønster (510), for at netværksenheden kan se adfærd på netværket, der afviger fra en normal godartet adfærd hos den netværksenhed, hvor netværksenheden er mindst én af en bruger og en indretning, der er tilknyttet et netværk;

10 identifikation af, om netværksenheden er i en brudtilstand i forhold til benchmarket for normal adfærd;

identifikation af, om brudtilstanden og en kæde af relevante adfærdsparametre, der afviger fra den normale godartede adfærd af den pågældende netværksenhed, svarer til en cybertrussel; og

15 afsendelse af en inokulationsmeddelelse med et inokulationsmønster, der beskriver brudtilstanden og kæden af relevante adfærdsparametre, til en målindretning for at advare om en potentiel cybertrussel, hvor et inokuleringsmodul anvender et kommunikationsmodul til at sende inokuleringsmeddelelsen til målindretningen på et andet netværk, der er beskyttet af et tilknyttet cybertrusselforsvarssystem via mindst én udgangsport, hvor inokuleringsmodulet bruger hændelsesdata, der beskriver brudtilstanden af en bruger eller indretning, der fungerer som netværksenhed, til at advare andre computerindretninger om potentielle cybertrusler, og hvor inokulationsmodulet anonymiserer inokulationsmønsteret for at fjerne enhver personlig identificerbar information for netværksenheden fra inokuleringsmønsteret.

2. Fremgangsmåde til cybertrusselforsvarssystemet (100) ifølge krav 1, yderligere omfattende:

30 at sende en afhjælpningshandlingsinstruktion i inokulationsmeddelelsen for at beskrive mindst én handling til at afhjælpe brudtilstanden.

**3.** Fremgangsmåde til cybertrusselforsvarssystemet (100) ifølge krav 1 eller 2, yderligere omfattende:

at anonymisere inokulationsmønsteret for at fjerne enhver personlig identificerbar information for netværksenheden fra inokuleringsmønsteret, og/eller

5 yderligere omfattende:

at præsentere en beskrivelse af brudtilstanden til en brugeranalytiker til gennemgang.

**4.** Fremgangsmåde til cybertrusselforsvarssystemet (100) ifølge krav 1, 2 eller 10 3, yderligere omfattende:

modtagelse, fra en brugeranalytiker, af mindst én af et udløsende input, der dirigerer transmissionen af inokulationsmeddelelsen til målindretningen, og et blokerende input, der forhindrer overførsel af inokulationsmeddelelsen til målindretningen.

15

**5.** Fremgangsmåde til cybertrusselforsvarssystemet (100) ifølge et hvilket som helst af de foregående krav, yderligere omfattende:

at generere en trusselsrisikoparameter med en liste over et sæt værdier, der beskriver aspekter af cybertruslen, eventuelt: yderligere omfattende:

20

at udfylde trusselsrisikoparameteren med mindst én af en konfidensscore (902), der angiver en trusselsandsynlighed, der beskriver en sandsynlighed for, at brudtilstanden er cybertruslen, en alvorlighedsscore (904), der angiver en procentdel af, at netværksenheden i brudtilstanden afviger fra den mindst ene model, og en konsekvensscore (906), der angiver en skadesgrad, der kan tilskrives cybertruslen.

25

**6.** Fremgangsmåde til cybertrusselforsvarssystemet (100) ifølge krav 5, yderligere omfattende:

at sammenligne trusselsrisikoparameteren med en benchmarkmatrix med et sæt benchmarkscores (1402, 1404, 1406) for at bestemme, om inokulationsmeddelelsen skal sendes, eventuelt:

30

yderligere omfattende:

at tildele en vægt til hver benchmarkscore for at tildele en relativ betydning til hver benchmarkscore.

5 **7.** Ikke-transitoriske computerlæsbart medium omfattende computerlæsbar kode, der kan betjenes, når den udføres af et eller flere behandlingsindretninger i cybertrusselforsvarssystemet, for at instruere en computerindretning til at udføre fremgangsmåden ifølge et hvilket som helst af de foregående krav.

10 **8.** Cybertrussel-kordinator-komponent til et cybertrusselforsvarssystem, omfattende:

et sammenligningsmodul, der er konfigureret til at udføre en sammenligning af inputdata, der overvåger en netværksenhed, med mindst én maskinlæringsmodel, der er trænet i en normal godartet adfærd hos netværksenheden ved hjælp af et benchmark for normal adfærd, der beskriver parametre, der svarer til et normalt aktivitetsmønster (510), for at netværksenheden kan opdage adfærd på netværket, der afviger fra en normal godartet adfærd hos den netværksenhed for at identificere, om netværksenheden er i en brudtilstand i forhold til benchmarket for normal adfærd, hvor netværksenheden repræsenterer mindst én af en bruger og en enhed forbundet med et netværk;

20 et cybertrusselsmodul, der er konfigureret til at identificere, om brudtilstanden, der er identificeret af sammenligningsmodulet, og en kæde af relevante adfærdsparametre, der afviger fra den normale godartede adfærd for den pågældende netværksenhed, svarer til en cybertrussel; og

25 et inokulationsmodul, der er konfigureret til at generere et inokulationsmønster, der beskriver brudtilstanden og kæden af relevante adfærdsparametre svarende til den cybertrussel, der er identificeret af cybertrusselsmodulet, og til at gemme inokuleringsmønsteret i en inokuleringsregistrering i en netværkstilgængelig inokulationsdatabase, hvor inokulationsmodulet er konfigureret til at bruge et kommunikationsmodul til at sende inokulationsmeddelelsen til en målindretning, på et andet netværk, der er beskyttet af et tilknyttet cybertrusselforsvarssystem, via mindst én udgangsport, hvor inokulationsmodulet er

30

konfigureret til at bruge hændelsesdata, der beskriver brudtilstanden af brugeren eller indretningen, der fungerer som netværksenheden, for at advare andre computerindretninger om potentielle cybertrusler, og hvor inokulationsmodulet er konfigureret til at anonymisere inokuleringsmønsteret for at fjerne enhver personlig identificerbar information for netværksenheden fra inokulationsmønsteret.

**9.** Cybertrussel-koordinatorkomponent til cybertrusselforsvarssystemet ifølge krav 8, hvor inokuleringsmodulet er konfigureret til at skabe en enhedsklynge til at gruppere netværksenheden med andre enheder i netværket baseret på kæden af relevante adfærdsparametre af inokulationsmønsteret, eventuelt: hvor inokulationsmodulet er konfigureret til at vælge målindretningen til notifikation vedrørende inokulationsmønsteret baseret på enhedsklyngen.

**10.** Cybertrussel-koordinatorkomponent til cybertrusselforsvarssystemet ifølge krav 8 eller 9, hvor inokulationsmodulet er konfigureret til at sende en inokuleringsrapport, der i det mindste angiver inokuleringsregistreringen til målindretningen relateret til netværksenheden om at instruere målindretningen for at hente inokuleringsregistreringen fra den netværkstilgængelige inokulationsdatabase.

**11.** Cybertrussel-koordinatorkomponent til cybertrusselforsvarssystemet ifølge et hvilket som helst af kravene 8 til 10, hvor inokuleringsregistreringen forbinder inokuleringsmønsteret med en afhjælpningshandlingsinstruktion for at beskrive mindst én handling til at afhjælpe cybertruslen, og/eller hvor inokulationsmodulet er konfigureret til at opdatere inokuleringsmønsteret i inokuleringsregistreringen baseret på en efterfølgende hændelse.

**12.** Cybertrussel-koordinatorkomponent til cybertrusselforsvarssystemet ifølge et hvilket som helst af kravene 8 til 11, hvor inokuleringsregistreringen forbinder inokuleringsmønsteret med et kontekstdatasæt, der beskriver mindst én af en netværksenhedshandling og en netværksenhedstilstand relateret til

cybertruslen.

- 5 **13.** Cybertrussel-koordinatorkomponent til cybertrusselforsvarssystemet ifølge et hvilket som helst af kravene 8 til 12, hvor inokuleringsregistreringen forbinder inokuleringsmønsteret med et eksternt datasæt, der er indsamlet fra mindst én datakilde uden for netværket og beskriver mindst én af en handling udefra og en ekstern tilstand relateret til cybertruslen.

# DRAWINGS

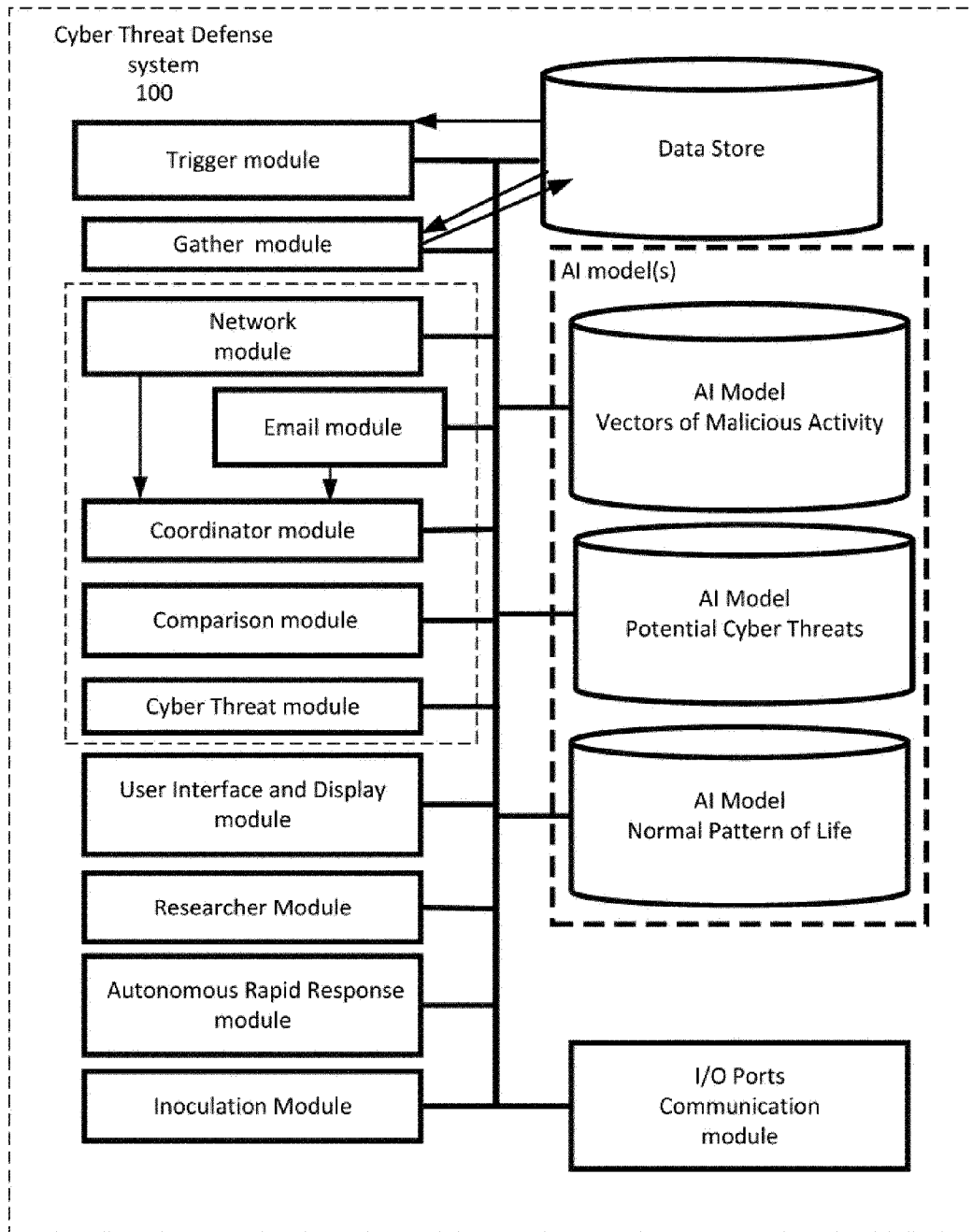


Fig. 1

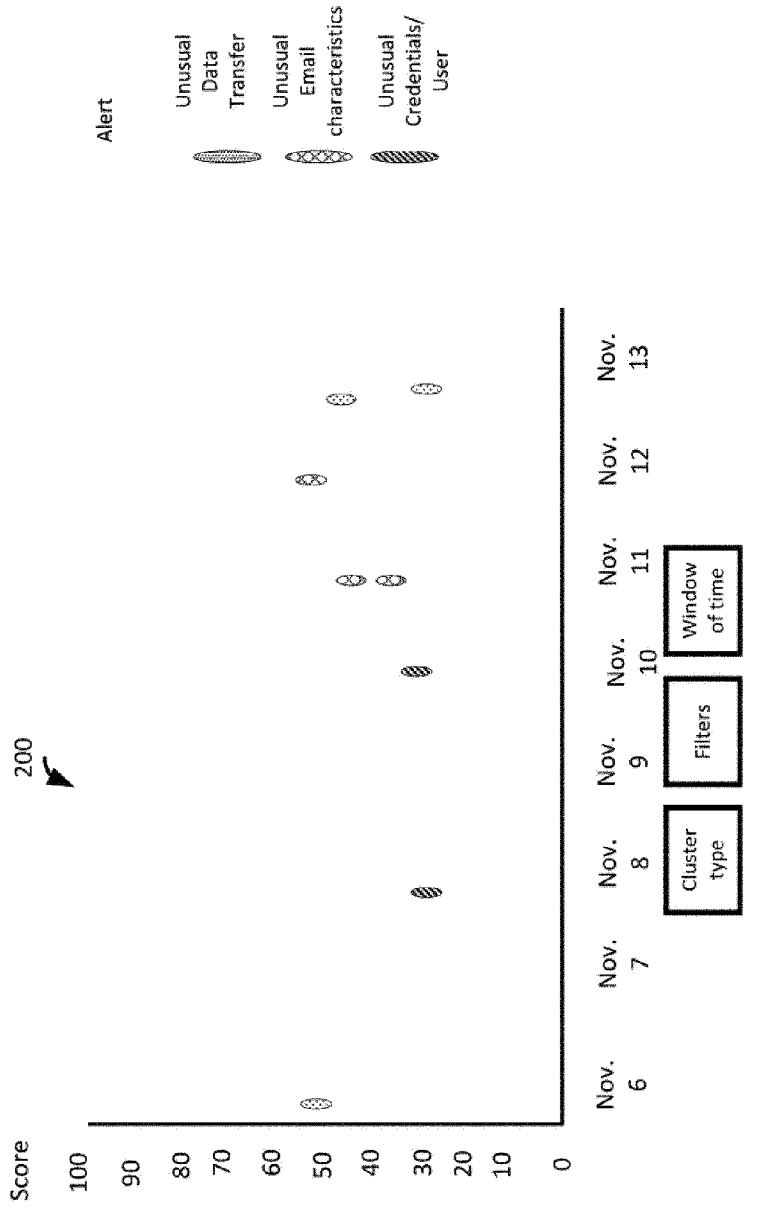


FIG. 2

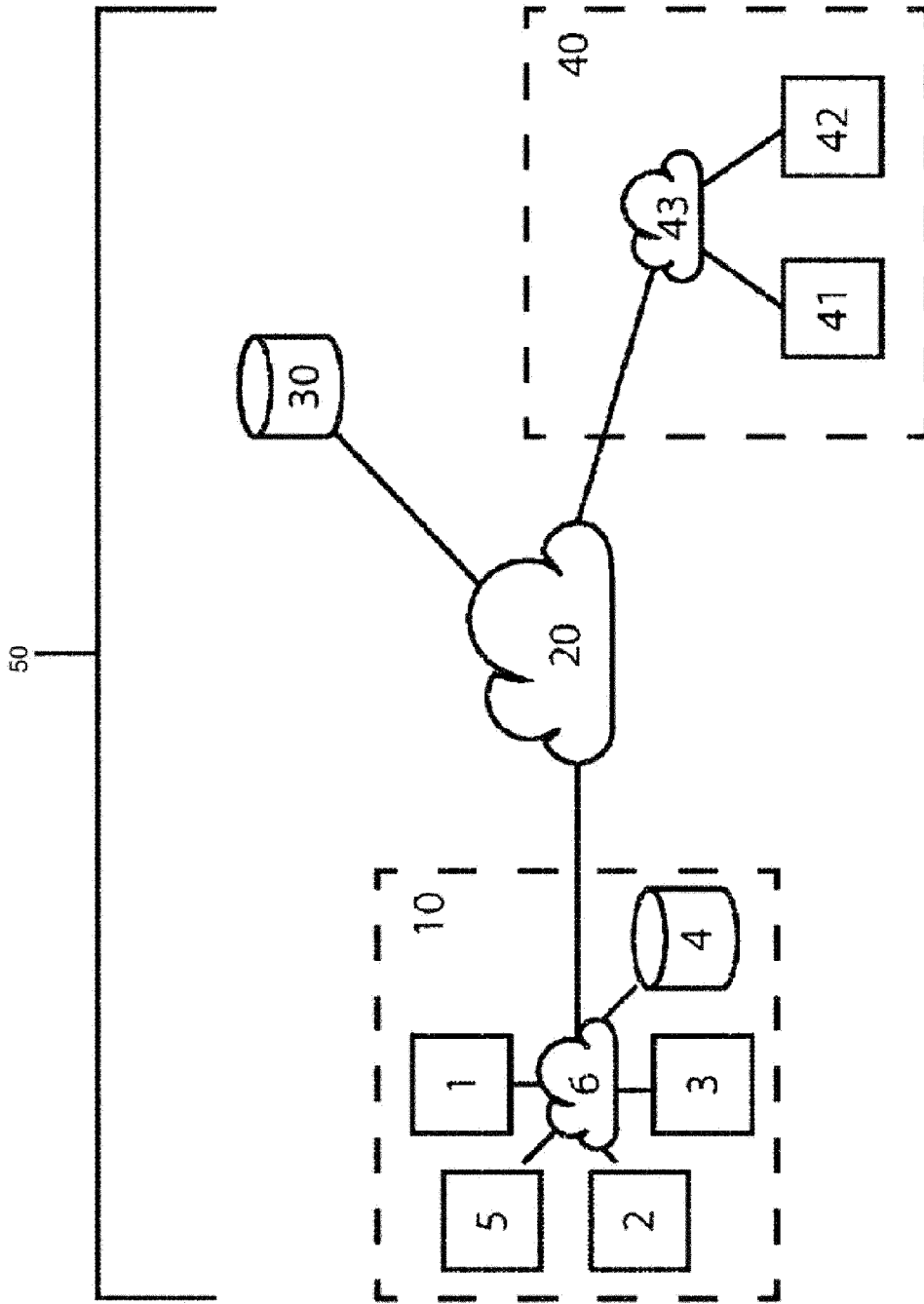


FIG. 3

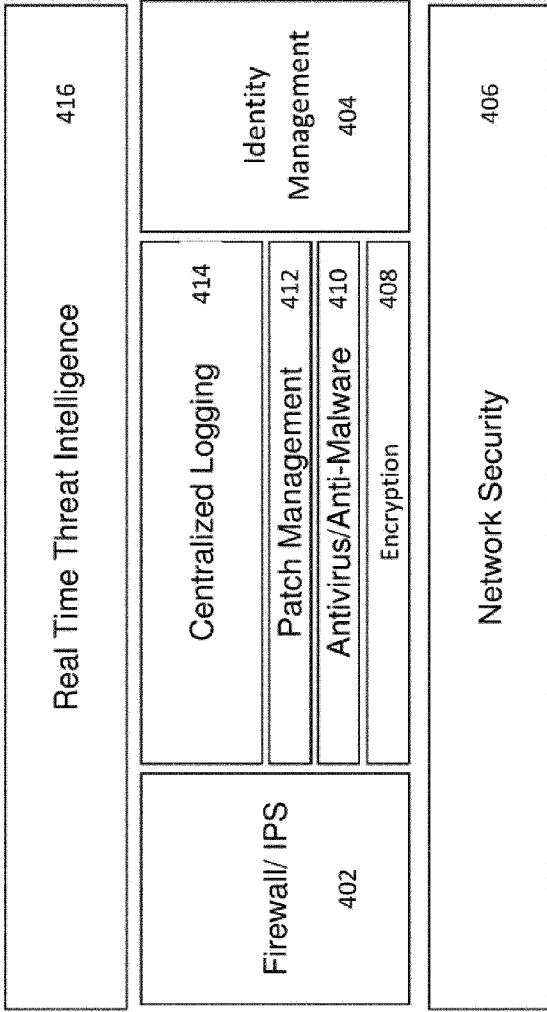


Fig. 4

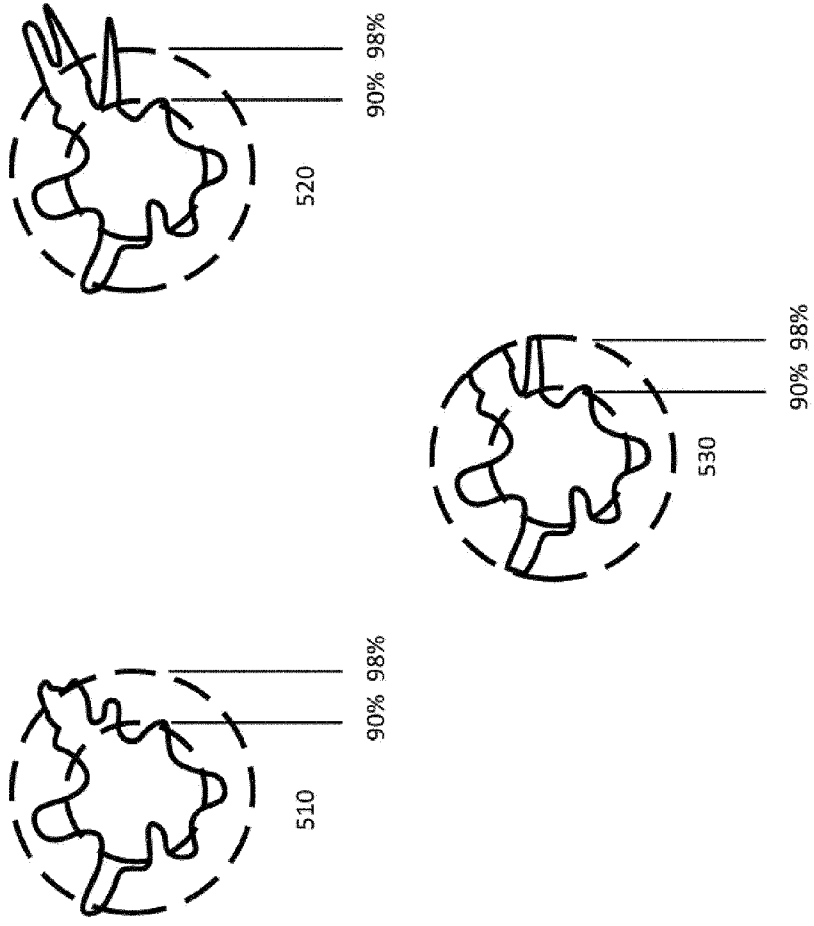


Fig. 5

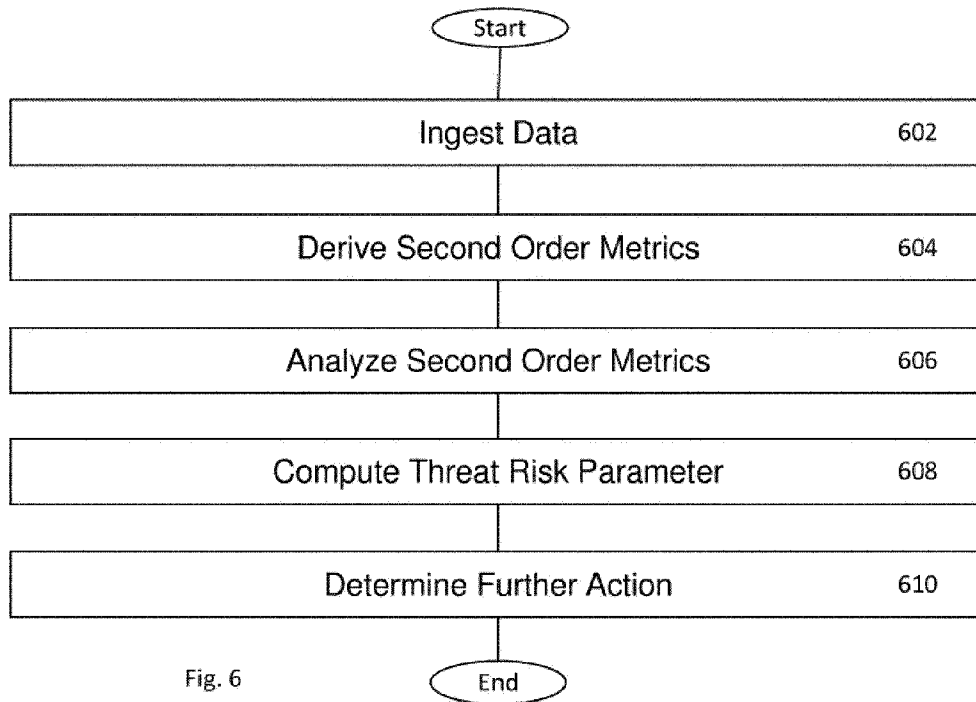


Fig. 6

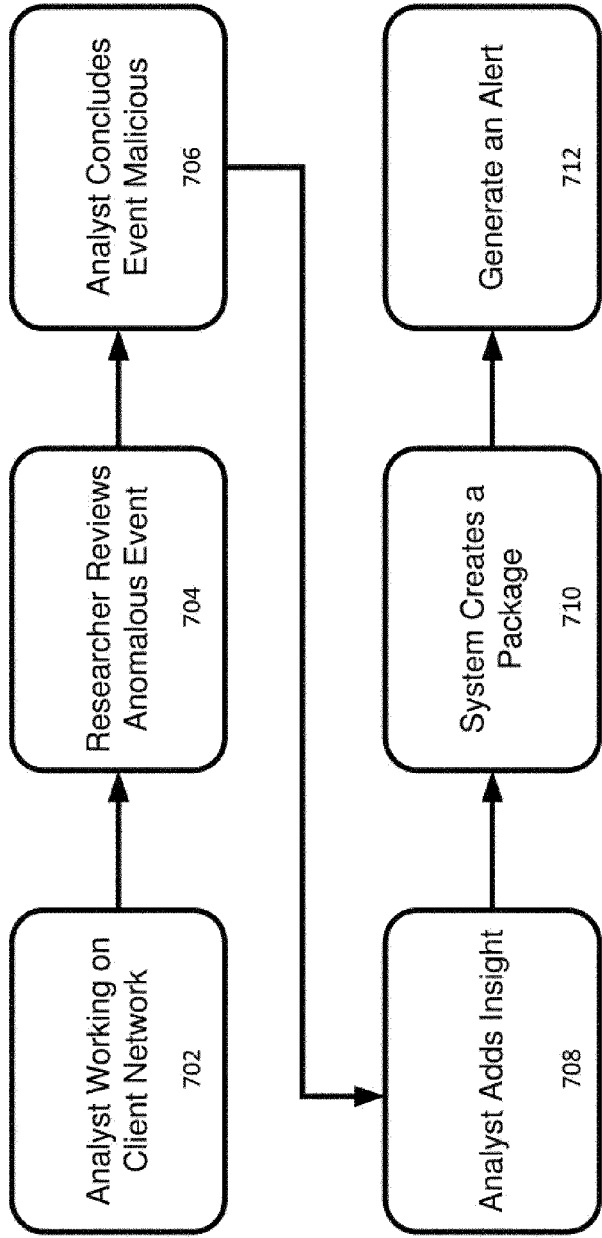


Fig. 7

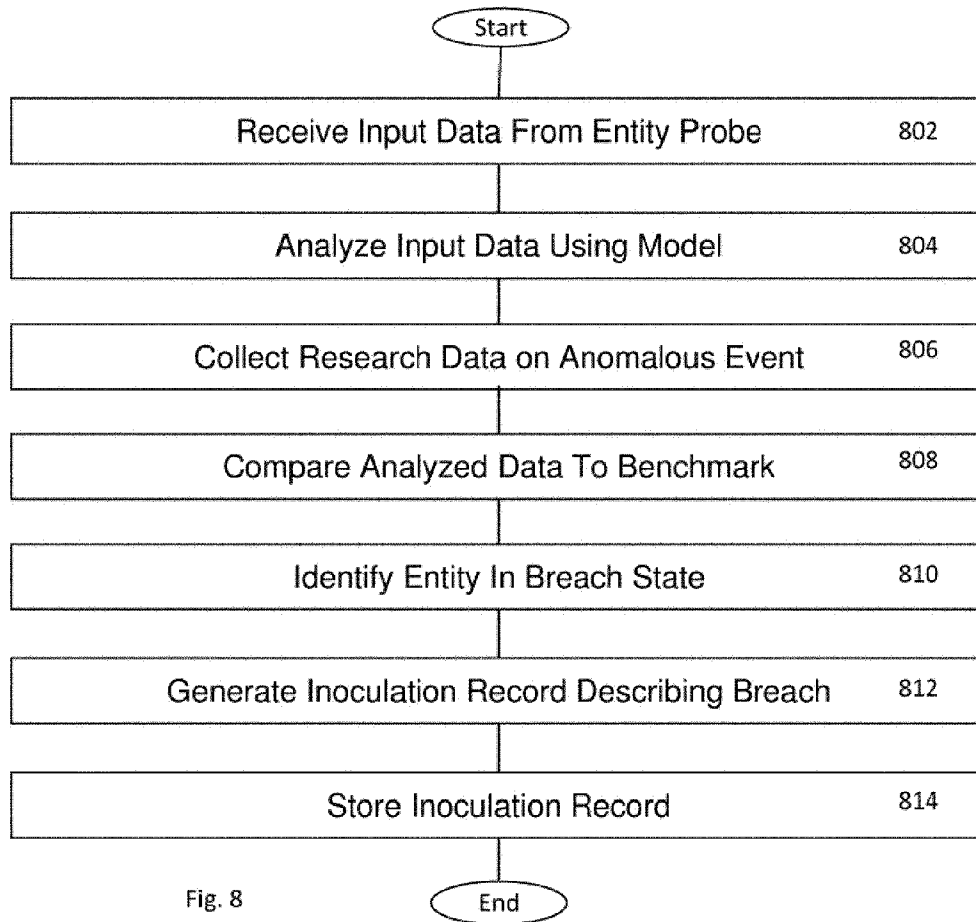


Fig. 8

Confidence Score 902	Severity Score 904	Consequence Score 906
----------------------------	-----------------------	-----------------------------

Fig. 9

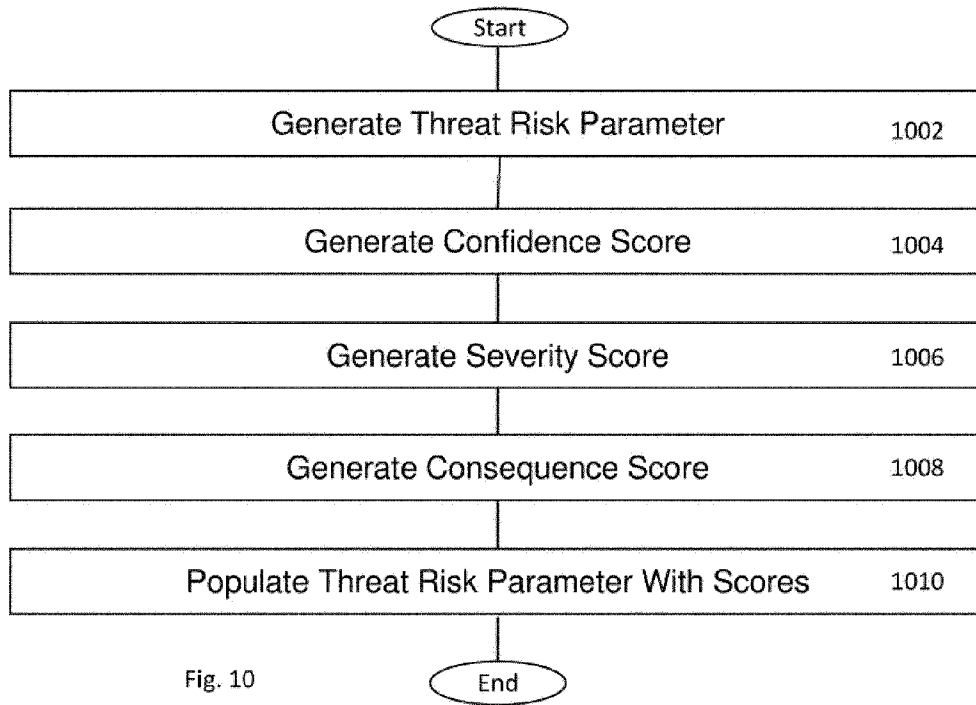
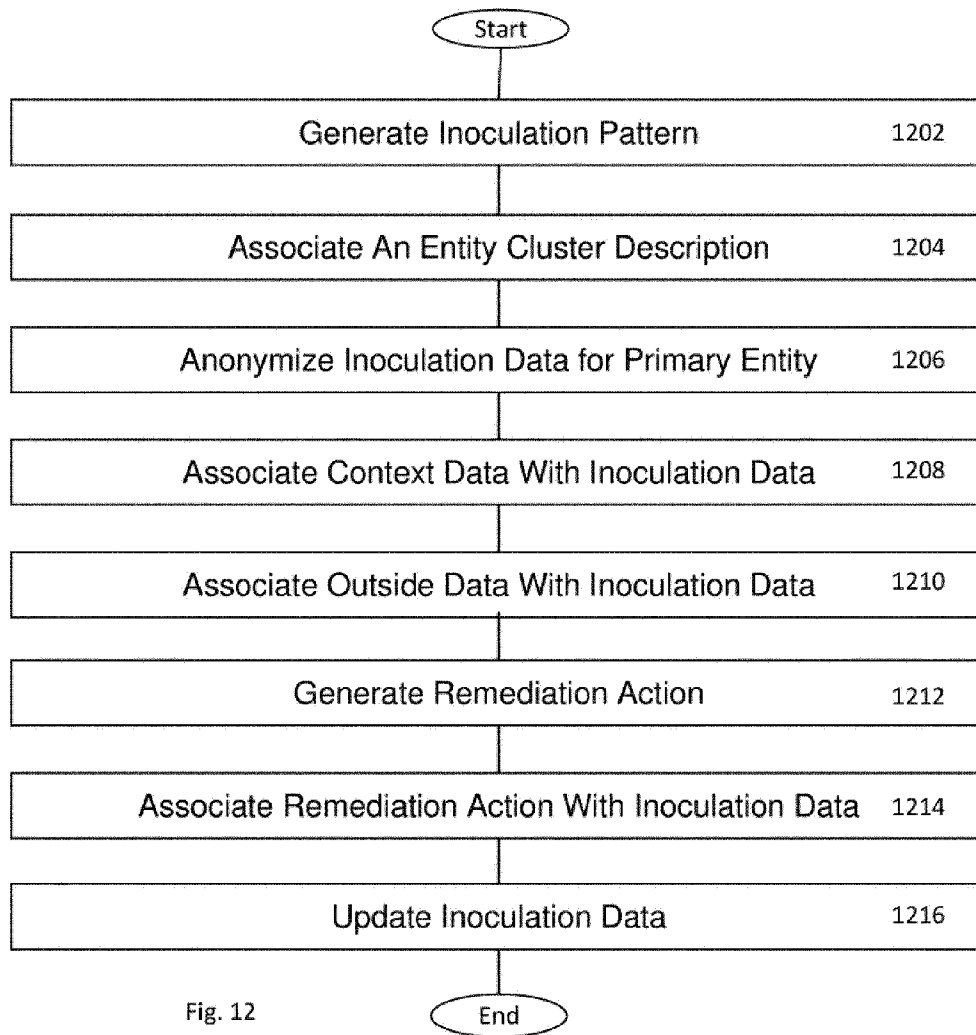


Fig. 10

Entity Cluster ID 1102	Inoculation Pattern 1104	Threat Risk Parameter 1106	Context Data 1108	Outside Data 1110	Remediation Action 1112
---------------------------	-----------------------------	-------------------------------	----------------------	----------------------	----------------------------

Fig. 11



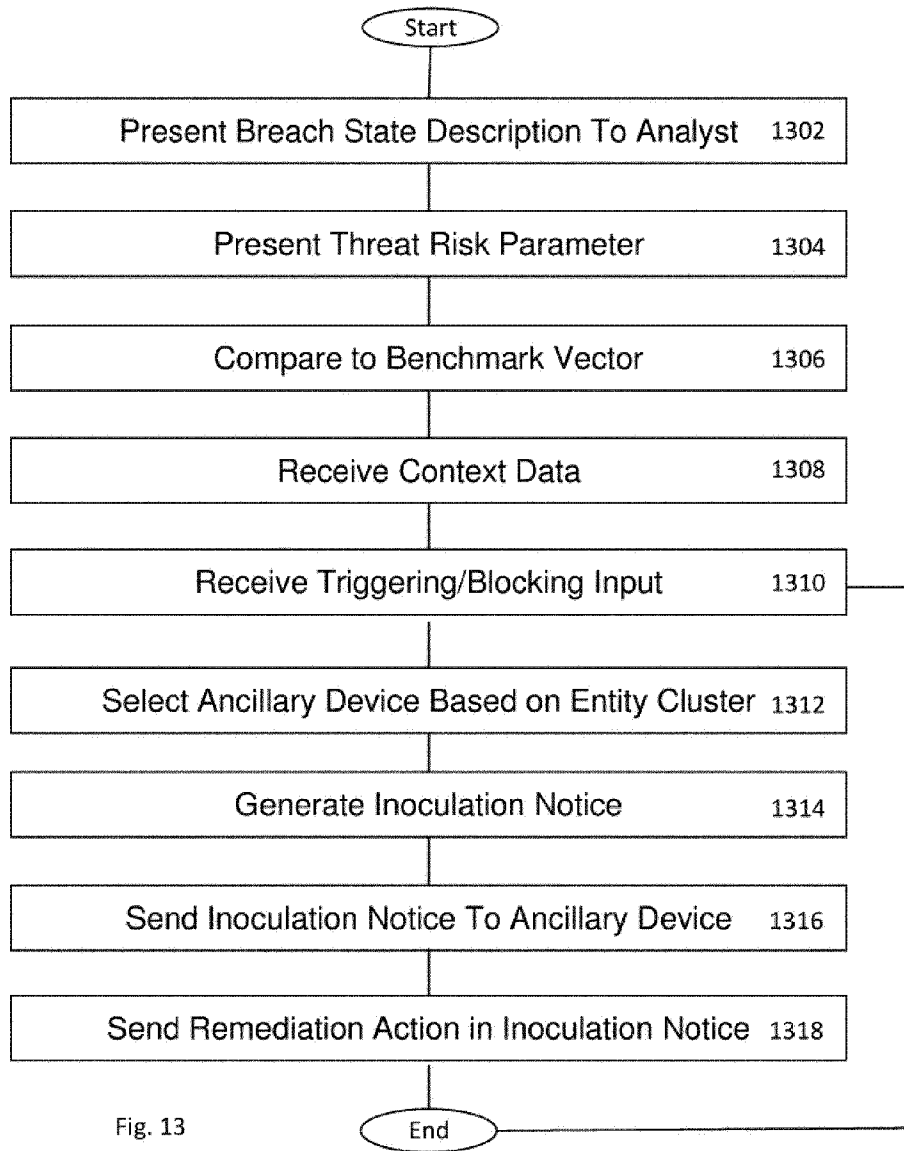


Fig. 13

Confidence Benchmark 1402	Severity Benchmark 1404	Consequence Benchmark 1406
Confidence Weight 1408	Severity Weight 1410	Consequence Weight 1412

Fig. 14

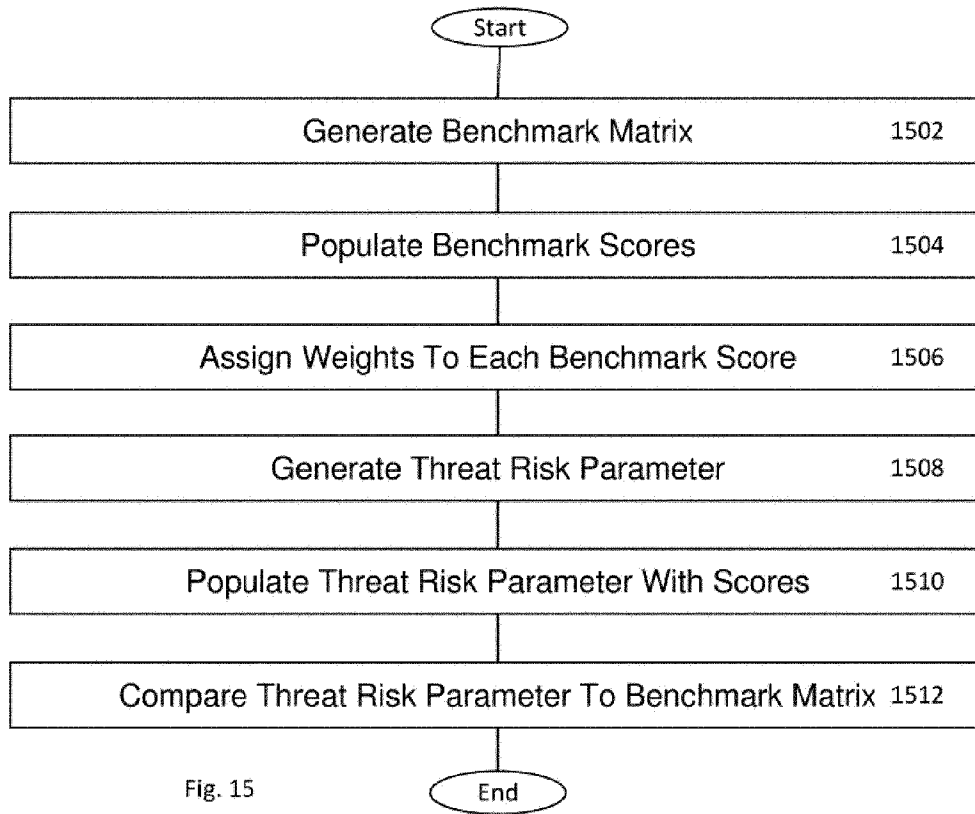


Fig. 15

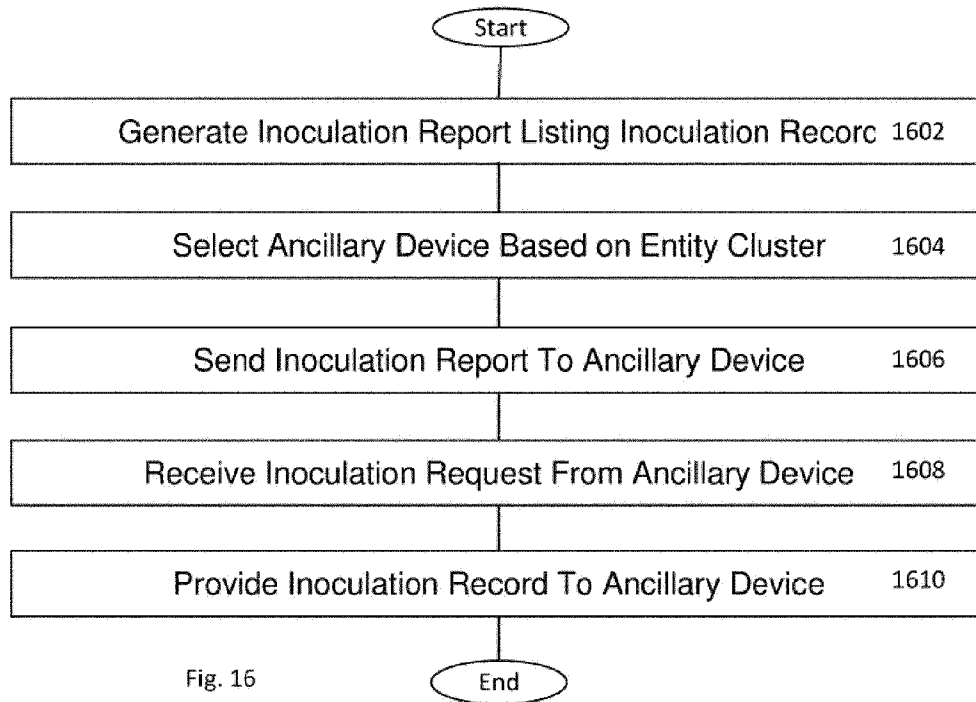


Fig. 16

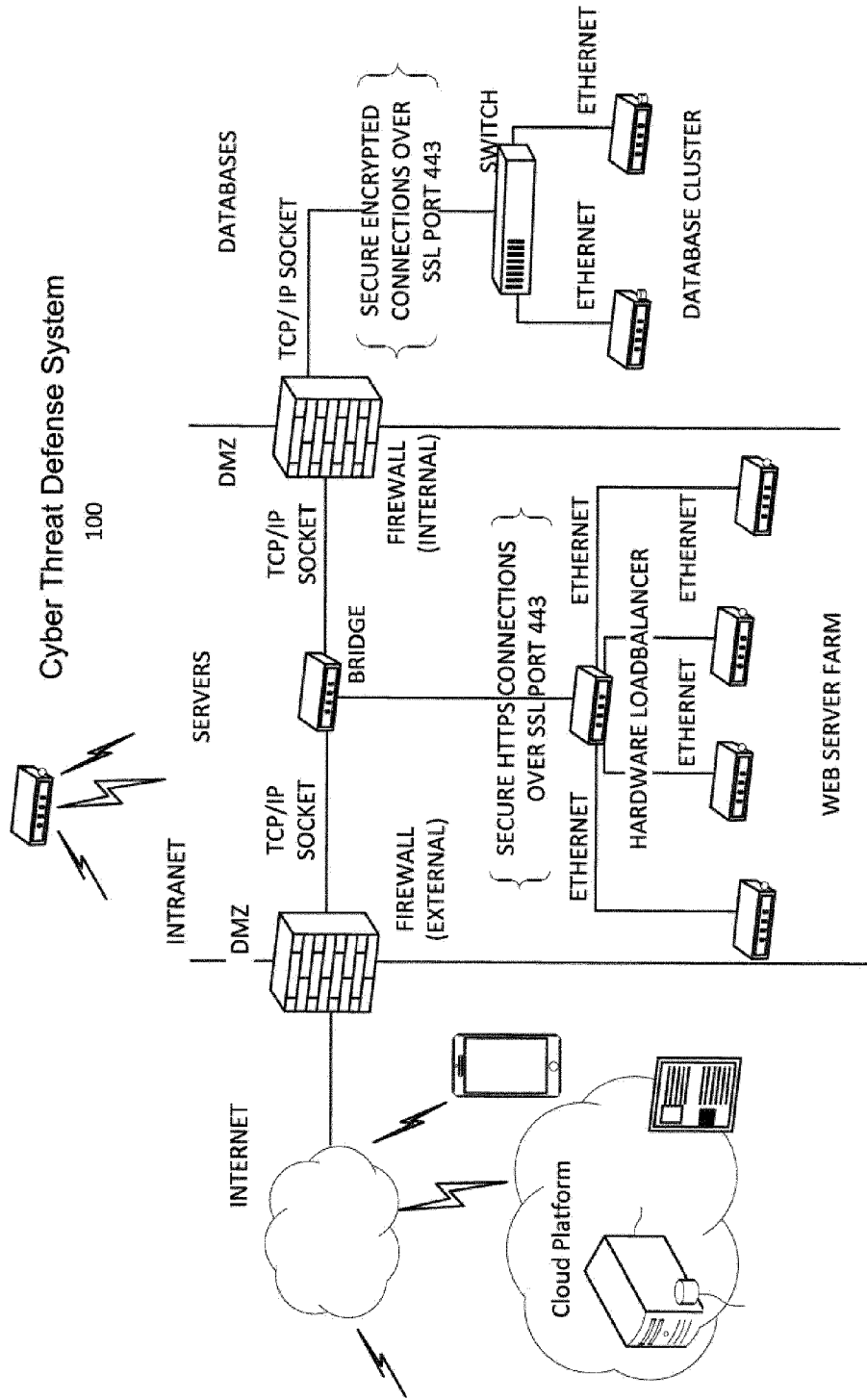


FIG. 17 Network