FORM 2

THE PATENTS ACT, 1970

(39 of 1970)

&

The Patent Rules, 2003

COMPLETE SPECIFICATION

(See sections 10 & rule 13)

1. TITLE OF THE INVENTION

CYBER ATTACK PREDICTION

2. APPLICANT (S)

NAME	NATIONALITY	ADDRESS
Chitkara University	IN	Chitkara University, Chandigarh-Patiala, National Highway, Village Jhansla, Rajpura, Punjab - 140401, India.
Bluest Mettle Solutions Private Limited	IN	ODC-4, Panchshil Tech Park, inside Courtyard by Marriott premises, Hinjewadi Phase - 1, Pune - 411057, Maharashtra, India.

3. PREAMBLE TO THE DESCRIPTION

COMPLETE SPECIFICATION

The following specification particularly describes the invention and the manner in which it is to be performed.

TECHNICAL FIELD

[0001] The present subject matter relates to the field of network security and more specifically, to systems and methods for managing network security of a computer network using machine learning techniques.

BACKGROUND

[0002] Cyberattacks have become a pervasive and persistent threat to organizations of all sizes and industries. Attackers continuously evolve their tactics, exploiting vulnerabilities in network systems and applications to gain unauthorized access, compromise data integrity, or disrupt critical operations. The consequences of successful cyberattacks can be severe, ranging from financial losses, reputational damage, regulatory non-compliance, to potential legal liabilities.

[0003] Conventional security measures, such as firewalls, intrusion detection systems, and access controls, play an essential role in safeguarding networks. However, they often rely on predetermined rules and patterns to detect and prevent threats. These rule-based systems have limitations in keeping up with the rapidly evolving nature of cyber threats. They may struggle to detect new attack vectors, recognize sophisticated intrusion techniques, or adapt to changing attack patterns.

SUMMARY

[0004] The present invention relates to a system and method for managing network security in a computer network. The system includes a processor, a security module, and a response module. The security module utilizes a machine learning model to generate risk scores for network transactions, providing an indication of the likelihood of a cyber attack. The response module initiates various actions based on the risk scores, including restricting communication for transactions exceeding a predetermined threshold and recording transactions while notifying a user device when the risk score falls within specified ranges.

[0005] Further, a method for managing network security in a computer network. The method involves generating risk scores for each network transaction

using a machine learning model, providing an indication of the likelihood of a cyber attack. Based on these risk scores, a variety of responses are initiated. When the risk score exceeds a predetermined threshold, communication of the device associated with the network transaction is restricted. Additionally, when the risk score is below the predetermined threshold but above a security notification threshold, the network transaction is recorded, and a user device is notified. This method enables proactive network security management, allowing for timely response and mitigation of potential cyber threats.

[0006] By employing this system and method, network security can be effectively managed, enabling proactive measures to prevent and mitigate potential cyber threats in the computer network.

BRIEF DESCRIPTION OF DRAWINGS

[0007] These and other features, aspects, and advantages of the present invention will become better understood when the following detailed description is read with reference to the accompanying drawings in which like characters represent like parts throughout the drawings, wherein:

[0008] FIG. 1 illustrates a device for managing network security of a computer network, according to one or more embodiments of the present disclosure; and

[0009] FIG. 2 illustrates a flowchart of a method for managing network security of a computer network, according to one or more embodiments of the present disclosure.

[0010] Further, skilled artisans will appreciate that elements in the drawings are illustrated for simplicity and may not have been necessarily been drawn to scale. For example, the flow charts illustrate the method in terms of the most prominent steps involved to help to improve understanding of aspects of the present invention. Furthermore, in terms of the construction of the device, one or more components of the device may have been represented in the drawings by conventional symbols, and the drawings may show only those specific details that are pertinent to understanding the embodiments of the present invention so as not to obscure the

drawings with details that will be readily apparent to those of ordinary skill in the art having benefit of the description herein.

DESCRIPTION OF EMBODIMENTS

[0011] The word “exemplary” or “embodiment” is used herein to mean “serving as an example, instance, or illustration.” Any implementation or aspect described herein as “exemplary” or as an “embodiment” is not necessarily to be construed as preferred or advantageous over other aspects of the disclosure. Likewise, the term “aspects” does not require that all aspects of the disclosure include the discussed feature, advantage, or mode of operation.

[0012] Embodiments will now be described in detail with reference to the accompanying drawings. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the aspects described herein. It will be apparent, however, to one skilled in the art, that these and other aspects may be practiced without some or all of these specific details. In addition, well known steps in a method of a process may be omitted from flow diagrams presented herein in order not to obscure the aspects of the disclosure. Similarly, well known components in a device may be omitted from figures and descriptions thereof presented herein in order not to obscure the aspects of the disclosure.

[0013] Before the present methods and systems are disclosed and described, it is to be understood that the methods and systems are not limited to specific methods, specific components, or to particular implementations. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting.

[0014] As used in the specification and the appended claims, the singular forms “a,” “an,” and “the” include plural referents unless the context clearly dictates otherwise. Ranges may be expressed herein as from “about” one particular value, and/or to “about” another particular value. When such a range is expressed, another embodiment includes from the one particular value and/or to the other particular value. Similarly, when values are expressed as approximations, by use of the antecedent “about,” it will be understood that the particular value forms another

embodiment. It will be further understood that the endpoints of each of the ranges are significant both in relation to the other endpoint, and independently of the other endpoint.

[0015] "Optional" or "optionally" means that the subsequently described event or circumstance may or may not occur, and that the description includes instances where said event or circumstance occurs and instances where it does not.

[0016] Throughout the description and claims of this specification, the word "comprise" and variations of the word, such as "comprising" and "comprises," means "including but not limited to," and is not intended to exclude, for example, other components, integers or steps. "Exemplary" means "an example of" and is not intended to convey an indication of a preferred or ideal embodiment. "Such as" is not used in a restrictive sense, but for explanatory purposes.

[0017] Disclosed are components that can be used to perform the disclosed methods and systems. These and other components are disclosed herein, and it is understood that when combinations, subsets, interactions, groups, etc. of these components are disclosed that while specific reference of each various individual and collective combinations and permutation of these may not be explicitly disclosed, each is specifically contemplated and described herein, for all methods and systems. This applies to all aspects of this application including, but not limited to, steps in disclosed methods. Thus, if there are a variety of additional steps that can be performed it is understood that each of these additional steps can be performed with any specific embodiment or combination of embodiments of the disclosed methods.

[0018] The present methods and systems may be understood more readily by reference to the following detailed description of preferred embodiments and the examples included therein and to the Figures and their previous and following description.

[0019] As will be appreciated by one skilled in the art, the methods and systems may take the form of an entirely hardware embodiment, an entirely software embodiment, or an embodiment combining software and hardware aspects. Furthermore, the methods and systems may take the form of a computer

program product on a computer-readable storage medium having computer-readable program instructions (e.g., computer software) embodied in the storage medium. More particularly, the present methods and systems may take the form of web-implemented computer software. Any suitable computer-readable storage medium may be utilized including hard disks, CD-ROMs, optical storage devices, or magnetic storage devices.

[0020] Embodiments of the methods and systems are described below with reference to block diagrams and flowchart illustrations of methods, systems, apparatuses and computer program products. It will be understood that each block of the block diagrams and flowchart illustrations, and combinations of blocks in the block diagrams and flowchart illustrations, respectively, can be implemented by computer program instructions. These computer program instructions may be loaded onto a general-purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions which execute on the computer or other programmable data processing apparatus create a means for implementing the functions specified in the flowchart block or blocks.

[0021] These computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including computer-readable instructions for implementing the function specified in the flowchart block or blocks. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer-implemented process such that the instructions that execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks.

[0022] Accordingly, blocks of the block diagrams and flowchart illustrations support combinations of means for performing the specified functions, combinations of steps for performing the specified functions and program

instruction means for performing the specified functions. It will also be understood that each block of the block diagrams and flowchart illustrations, and combinations of blocks in the block diagrams and flowchart illustrations, can be implemented by special purpose hardware-based computer systems that perform the specified functions or steps, or combinations of special purpose hardware and computer instructions.

[0023] The foregoing objects of the invention are accomplished and the problems and shortcomings associated with the prior art techniques and approaches are overcome by the present invention as described below in the preferred embodiment.

[0024] The infrastructure and sensitive data of companies are at substantial danger as a result of the increasing frequency and sophistication of cyberattacks. Conventional security measures sometimes rely on rule-based systems that are incapable of changing to meet the needs of emerging threats.

[0025] By utilising both historical and current data, machine learning (ML) systems have shown impressive skills in identifying and thwarting cyberattacks. This patent describes a system and method for using machine learning to predict and prevent cyber attacks. The system includes an ML model trained on a large dataset of cyber threat intelligence, which continuously learns and adapts to new data. The system uses this ML model to analyze network traffic, detect anomalies, and identify potential attacks. The system also includes a response module that can take proactive measures to prevent or mitigate an attack, such as blocking traffic or alerting security personnel.

[0026] FIG. 1 illustrates a system 100 for managing network security of a computer network. In an example, the system 100 comprises a processor 102, a security module 104, and a response module 106.

[0027] According to an embodiment of the present subject matter, the security module 104, coupled to the processor 102, generates risk scores for network transactions occurring within the computer network. In said embodiment, the security module 104 analyzes various factors and characteristics of the network transactions based on a machine learning model to determine the likelihood of a

cyber attack. Examples of the machine learning model which may be implemented may include, but are not limited to, supervised learning models, unsupervised learning based anomaly detection models, deep learning models, or a combination thereof.

[0028] In an implementation the machine learning model may consider factors such as the source and destination IP addresses, the type of network protocol used, the size and frequency of data packets, the behavior patterns of the devices involved, and any known vulnerabilities or threat indicators. As an example, the model can analyze the IP addresses involved in the network transaction. For example, if the source IP address is known to be associated with malicious activities or if the destination IP address is flagged as suspicious, the risk score may be higher. In another example, the type of network protocol used in the transaction can provide insights into potential vulnerabilities. For instance, certain protocols may be more prone to attacks or have known security weaknesses, leading to a higher risk score. In yet another example, unusually large or frequent data packets can indicate potential data exfiltration or malicious activities. The model can take into account these factors to assess the risk associated with the transaction. In a further example, by analyzing the historical behavior patterns of devices involved in the network transaction, the model can identify anomalies or deviations from normal behavior. Suspicious or abnormal behavior can contribute to a higher risk score. Further, the model can incorporate information about known vulnerabilities or threat indicators, such as recently discovered malware or attack patterns. If the network transaction exhibits characteristics associated with these threats, the risk score may be increased.

[0029] Based on this analysis, the security module 104 assigns a risk score to each network transaction. The risk score provides an indication of the potential security risk associated with that particular transaction. Higher risk scores suggest a higher likelihood of a cyber attack, while lower risk scores indicate a lower likelihood.

[0030] Thus, the security module 104 generates risk scores that provide an indication of the potential security risk associated with each network transaction, as

described above.

[0031] In an embodiment, the response module 106, coupled to the processor 102, initiates a range of responses based on the risk scores assigned to each network transaction. These responses aim to mitigate potential security threats and ensure the overall network security.

[0032] For instance, when the risk score for a network transaction exceeds a predetermined threshold, the response module 106 triggers a response that restricts communication of the device associated with that transaction. This restriction helps prevent further potential cyber attacks and safeguards the network from potential harm.

[0033] Alternatively, when the risk score for a network transaction is below the predetermined threshold but still above a security notification threshold, the response module 106 initiates a different response. In this case, the response module 106 records the network transaction for future reference and notifies a user device. This notification serves as a security alert, informing the user about the potential security risks associated with the specific network transaction.

[0034] Furthermore, in an embodiment, the system includes a data ingestion module 108 and a training module 110, both coupled to the processor 102. In said embodiment, the data ingestion module 108 is coupled to the processor 102 and is responsible for providing network data to a machine learning (ML) model. In an example, the network data comprises various types of data, including but not limited to network logs, firewall data, and data from one or more intrusion detection systems. The data ingestion module 108 ensures the collection and preprocessing of the network data, making it suitable for training the ML model.

[0035] Furthermore, in said embodiment, the training module 110is configured for training the ML model based on the network data. By leveraging machine learning techniques, the ML model can learn patterns and identify anomalies within the network data, aiding in the detection and prevention of potential security threats. In operation, the data ingestion module 108 continuously gathers network data from various sources and feeds it to the ML model. The ML model, trained using the training module 110, processes the network data to identify

and classify potential security threats, unauthorized access attempts, or abnormal network behavior.

[0036] Furthermore, in an embodiment, the data ingestion module 108 is configured to receive cyber threat data from an external entity, such as government bodies or business associations. This allows the system to stay up-to-date with the latest threat information and enhance the accuracy of threat detection. In said embodiment, the training module 110 is further configured to update the ML model based on the received cyber threat data, ensuring that the system adapts to evolving threats and maintains optimal performance.

[0037] Yet further, in an embodiment, the system 100 comprises a feedback module 112 coupled to the processor 102. The feedback module 112 gathers feedback data that indicates the performance of the system. This feedback data can include metrics, performance indicators, or user feedback. In said embodiment, the training module 110 utilizes this feedback data to further train and optimize the ML model, improving the system 100's performance over time.

[0038] Furthermore, in an embodiment, the security module 104 may generate a comprehensive security report. The security report provides information about security risks, the number of security attacks, and the corresponding measures implemented to mitigate those risks. This report serves as a valuable resource for monitoring and evaluating the effectiveness of the security measures implemented within the network.

[0039] FIG. 2 illustrates a method 200 for managing network security of a computer network. In an example, the method 200 is performed by the system 100.

[0040] At step 202, the method 200 includes, generating a risk score for each network transaction occurring in the computer network using a machine learning model. This step involves applying a trained machine learning model to evaluate the risk associated with individual network transactions. During this step, the method utilizes the trained machine learning model, which has learned patterns and anomalies from the collected and preprocessed network data. For each network transaction that takes place within the computer network, the machine learning model assesses various characteristics, such as the source, destination, type of

communication, and other relevant features. Based on the learned patterns and anomalies, the machine learning model calculates a risk score that represents the level of potential risk associated with each network transaction.

[0041] The risk score provides a measure of the likelihood or severity of a security threat or anomaly within the network transaction. Higher risk scores indicate a higher potential risk, while lower scores suggest a lower likelihood of a security threat. The generation of risk scores for each network transaction provides valuable insight into the security posture of the computer network. It allows the identification of potentially suspicious or malicious activities that may pose a risk to the network's integrity or data security. These risk scores obtained from the method assist in subsequent actions, such as triggering alerts, initiating security measures, or conducting further analysis by security personnel. They help prioritize responses and allocate resources effectively to address the most critical security threats and anomalies within the computer network.

[0042] At step 204, the method 200 includes initiating a plurality of responses based on the risk score for each network transaction. In an example, the response includes restricting communication of a device associated with the network transaction when the risk score exceeds a predetermined threshold. If the risk score calculated for a particular network transaction exceeds a predetermined threshold, indicating a high level of potential risk, the method initiates the restriction of communication for the device associated with that transaction. This restriction aims to isolate the potentially compromised device or prevent further malicious activities from spreading within the network.

[0043] In another example, the response includes recording the network transaction and notifying a user device when the risk score is below the predetermined threshold and above a security notification threshold. If the risk score for a network transaction falls below the predetermined threshold but still surpasses a security notification threshold, the method proceeds to record the transaction for future analysis and notifies a user device. This notification alerts the user or network administrator about the transaction, indicating that it has been flagged as potentially suspicious or requiring attention.

[0044] Furthermore, in an embodiment, the method includes a step in which network data is provided to the machine learning model. This includes various types of data such as network logs, firewall data, and data from intrusion detection systems. By feeding this comprehensive network data to the model, it is trained to learn patterns and anomalies that aid in accurate threat detection and analysis.

[0045] Furthermore, in an embodiment, the method includes a step in which the machine learning model is trained based on the network data. This training process enhances the model's ability to identify security threats within the network data by continuously updating its knowledge and adapting to evolving threat landscapes.

[0046] Furthermore, in an embodiment, the method includes a step in which cyber threat data from an external entity is recieved. This additional data, received from sources like threat intelligence feeds or security service providers, enriches the knowledge of the machine learning model, enabling it to stay up to date with emerging threats and improving its accuracy in detecting and mitigating potential security risks.

[0047] Furthermore, in an embodiment, the method includes a step in which the machine learning model is updated using the received cyber threat data. This ensures that the model remains current and capable of effectively responding to new and evolving threats within the network.

[0048] Furthermore, in an embodiment, the method includes a step in which feedback data indicative of the system's performance is gathered. This feedback may include metrics, performance indicators, or user feedback that provide valuable insights into the effectiveness of the system. Furthermore, in an embodiment, the method includes a step in which the machine learning model is further trained based on the gathered feedback data. By incorporating this feedback, the model can adapt and improve its performance over time, enhancing the accuracy and effectiveness of threat detection within the network.

[0049] Furthermore, in an embodiment, the method includes a step in which a security report is generated. This report contains comprehensive information about the security risks identified within the network, the number of security attacks

detected, and the corresponding measures implemented to address them. The security report facilitates effective risk management, decision-making, and provides valuable insights into the overall security posture of the network.

[0050] The present disclosure also provides for various advancements and capabilities in managing network security using machine learning. These include:

- Anomaly detection: Machine learning algorithms can learn the normal patterns of network behavior and identify anomalies or deviations from these patterns. This enables the system to detect previously unseen cyber threats and flag them for further investigation.
- Real-time monitoring: Machine learning models can continuously analyze network traffic and system logs in real-time, allowing for immediate detection and response to cyber attacks. This proactive approach helps to minimize the impact of attacks and prevent further damage.
- Adaptive learning: Machine learning models can adapt and evolve based on new data and emerging threat patterns. As attackers develop new techniques, the system can learn from these changes and update its algorithms to stay ahead of evolving threats.
- Behavioral analysis: Machine learning algorithms can analyze user behavior and identify abnormal activities that may indicate a compromised account or malicious intent. By monitoring user behavior, the system can provide an additional layer of protection against unauthorized access.
- Automated response: Machine learning-based systems can be integrated with automated response mechanisms to take immediate action against detected threats. This can include blocking suspicious IP addresses, quarantining compromised devices, or triggering alerts for security personnel to investigate.
- Reduced false positives: Traditional rule-based systems often generate a large number of false positive alerts, leading to alert fatigue and decreased efficiency. Machine learning can significantly reduce false positives by accurately distinguishing between genuine threats and benign activities, improving the overall efficiency of incident response teams.

- Threat intelligence integration: Machine learning systems can leverage threat intelligence feeds and data from various sources to enhance their predictive capabilities. By analyzing historical attack data and incorporating external threat intelligence, the system can identify emerging attack patterns and proactively defend against them.
- Scalability: Machine learning models can handle large volumes of data and scale efficiently to monitor and protect complex network environments. Whether it's a small business or a large enterprise, machine learning-based cybersecurity systems can adapt to the size and complexity of the organization's infrastructure.
- Continuous learning: Machine learning algorithms can continuously learn from new data and adapt their models accordingly. This enables the system to improve its accuracy and detection capabilities over time, as it gains more insights from ongoing cyber threats and attack patterns.
- Visualization and reporting: Machine learning systems can provide intuitive visualizations and comprehensive reports that highlight security risks, detected threats, and overall system health. These visual representations help security analysts and decision-makers understand the security posture of their organization and make informed decisions.
- Deep learning capabilities: Deep learning, a subset of machine learning, utilizes artificial neural networks with multiple layers to process complex data and extract intricate patterns. By employing deep learning algorithms, cybersecurity systems can detect sophisticated and previously unknown attack techniques, such as polymorphic malware or advanced persistent threats (APTs).
- Ensemble learning: Ensemble learning combines multiple machine learning models to improve prediction accuracy and robustness. By integrating diverse algorithms and leveraging their collective intelligence, the system can make more reliable predictions and effectively detect and prevent cyber attacks.
- Unsupervised learning: Unsupervised learning techniques allow the system

to learn from unlabeled data, making it capable of detecting anomalies and identifying unknown threats without relying on predefined rules or labeled datasets. This enables the system to adapt to emerging threats that have not been encountered before.

- Transfer learning: Transfer learning enables the system to leverage knowledge gained from one task or domain to improve performance on another task or domain. In the context of cybersecurity, transfer learning can be used to transfer knowledge learned from one organization's security data to another, enhancing the effectiveness of threat detection and prevention.
- Explainable AI: Explainable AI techniques aim to provide transparency and interpretability to machine learning models. By understanding how the system arrives at its decisions, security analysts can better trust and validate the predictions

[0051] The system and method disclosed herein offer several advantages. By combining network data analysis, threat detection, performance optimization, and security reporting, the system provides a holistic approach to network security. The utilization of machine learning techniques enhances the accuracy and efficiency of threat detection, ensuring prompt identification and prevention of potential security threats.

[0052] In an illustrative configuration, the system 100, may be a computing device, such as a server or a workstation desktop, and may include one or more processors, such as the processor 102, one or more memory devices (generically referred to herein as memory), one or more input/output (I/O) interface(s), one or more network interface(s), one or more sensors or sensor interface(s), one or more transceivers, one or more optional speakers, one or more optional microphones, and data storage. The computing device may further include one or more buses that functionally couple various components of the remote server. The computing device may further include one or more antenna(e) 1034 that may include, without limitation, a cellular antenna for transmitting or receiving signals to/from a cellular network infrastructure, an antenna for transmitting or receiving Wi-Fi signals to/from an access point (AP), a Global Navigation Satellite System (GNSS) antenna

for receiving GNSS signals from a GNSS satellite, a Bluetooth antenna for transmitting or receiving Bluetooth signals, a Near Field Communication (NFC) antenna for transmitting or receiving NFC signals, and so forth. These various components will be described in more detail hereinafter.

[0053] The bus(es) may include at least one of a system bus, a memory bus, an address bus, or a message bus, and may permit exchange of information (e.g., data (including computer-executable code), signaling, etc.) between various components of the remote server. The bus(es) may include, without limitation, a memory bus or a memory controller, a peripheral bus, an accelerated graphics port, and so forth. The bus(es) may be associated with any suitable bus architecture including, without limitation, an Industry Standard Architecture (ISA), a Micro Channel Architecture (MCA), an Enhanced ISA (EISA), a Video Electronics Standards Association (VESA) architecture, an Accelerated Graphics Port (AGP) architecture, a Peripheral Component Interconnects (PCI) architecture, a PCI-Express architecture, a Personal Computer Memory Card International Association (PCMCIA) architecture, a Universal Serial Bus (USB) architecture, and so forth.

[0054] The memory of the computing device may include volatile memory (memory that maintains its state when supplied with power) such as random access memory (RAM) and/or non-volatile memory (memory that maintains its state even when not supplied with power) such as read-only memory (ROM), flash memory, ferroelectric RAM (FRAM), and so forth. Persistent data storage, as that term is used herein, may include non-volatile memory. In certain example embodiments, volatile memory may enable faster read/write access than non-volatile memory. However, in certain other example embodiments, certain types of non-volatile memory (e.g., FRAM) may enable faster read/write access than certain types of volatile memory.

[0055] In various implementations, the memory may include multiple different types of memory such as various types of static random access memory (SRAM), various types of dynamic random access memory (DRAM), various types of unalterable ROM, and/or writeable variants of ROM such as electrically erasable programmable read-only memory (EEPROM), flash memory, and so forth. The

memory 1004 may include main memory as well as various forms of cache memory such as instruction cache(s), data cache(s), translation lookaside buffer(s) (TLBs), and so forth. Further, cache memory such as a data cache may be a multi-level cache organized as a hierarchy of one or more cache levels (L1, L2, etc.).

[0056] The data storage may include removable storage and/or non-removable storage including, but not limited to, magnetic storage, optical disk storage, and/or tape storage. The data storage may provide non-volatile storage of computer-executable instructions and other data. The memory and the data storage, removable and/or non-removable, are examples of computer-readable storage media (CRSM) as that term is used herein.

[0057] The data storage may store computer-executable code, instructions, or the like that may be loadable into the memory and executable by the processor(s) to cause the processor(s) to perform or initiate various operations. The data storage may additionally store data that may be copied to memory for use by the processor(s) during the execution of the computer-executable instructions. Moreover, output data generated as a result of execution of the computer-executable instructions by the processor(s) may be stored initially in memory, and may ultimately be copied to data storage for non-volatile storage.

[0058] More specifically, the data storage may store one or more operating systems (O/S); one or more database management systems (DBMS); and one or more program module(s), applications, engines, computer-executable code, scripts, or the like such as, for example, one or more machine learning module(s), one or more communication module(s), one or more content scanning module(s), and/or one or more prediction module(s). Some or all of these module(s) may be sub-module(s). Any of the components depicted as being stored in data storage may include any combination of software, firmware, and/or hardware. The software and/or firmware may include computer-executable code, instructions, or the like that may be loaded into the memory for execution by one or more of the processor(s). Any of the components depicted as being stored in data storage may support functionality described in reference to correspondingly named components earlier in this disclosure.

[0059] The data storage may further store various types of data utilized by components of the computing device. Any data stored in the data storage may be loaded into the memory for use by the processor(s) in executing computer-executable code. In addition, any data depicted as being stored in the data storage may potentially be stored in one or more datastore(s) and may be accessed via the DBMS and loaded in the memory for use by the processor(s) in executing computer-executable code. The datastore(s) may include, but are not limited to, databases (e.g., relational, object-oriented, etc.), file systems, flat files, distributed datastores in which data is stored on more than one node of a computer network, peer-to-peer network datastores, or the like. The datastore(s) may include, for example, purchase history information, user action information, user profile information, a database linking search queries and user actions, and other information.

[0060] The processor(s) 102 may be configured to access the memory and execute computer-executable instructions loaded therein. For example, the processor(s) may be configured to execute computer-executable instructions of the various program module(s), applications, engines, or the like of the computing device to cause or facilitate various operations to be performed in accordance with one or more embodiments of the disclosure. The processor(s) may include any suitable processing unit capable of accepting data as input, processing the input data in accordance with stored computer-executable instructions, and generating output data. The processor(s) may include any type of suitable processing unit including, but not limited to, a central processing unit, a microprocessor, a Reduced Instruction Set Computer (RISC) microprocessor, a Complex Instruction Set Computer (CISC) microprocessor, a microcontroller, an Application Specific Integrated Circuit (ASIC), a Field-Programmable Gate Array (FPGA), a System-on-a-Chip (SoC), a digital signal processor (DSP), and so forth. Further, the processor(s) may have any suitable microarchitecture design that includes any number of constituent components such as, for example, registers, multiplexers, arithmetic logic units, cache controllers for controlling read/write operations to cache memory, branch predictors, or the like. The microarchitecture design of the processor(s) may be

capable of supporting any of a variety of instruction sets.

[0061] It should further be appreciated that the system may include alternate and/or additional hardware, software, or firmware components beyond those described or depicted without departing from the scope of the disclosure. More particularly, it should be appreciated that software, firmware, or hardware components depicted as forming part of the computing device are merely illustrative and that some components may not be present or additional components may be provided in various embodiments. While various illustrative program module(s) have been depicted and described as software module(s) stored in data storage, it should be appreciated that functionality described as being supported by the program module(s) may be enabled by any combination of hardware, software, and/or firmware. It should further be appreciated that each of the above-mentioned module(s) may, in various embodiments, represent a logical partitioning of supported functionality. This logical partitioning is depicted for ease of explanation of the functionality and may not be representative of the structure of software, hardware, and/or firmware for implementing the functionality. Accordingly, it should be appreciated that functionality described as being provided by a particular module may, in various embodiments, be provided at least in part by one or more other module(s). Further, one or more depicted module(s) may not be present in certain embodiments, while in other embodiments, additional module(s) not depicted may be present and may support at least a portion of the described functionality and/or additional functionality. Moreover, while certain module(s) may be depicted and described as sub-module(s) of another module, in certain embodiments, such module(s) may be provided as independent module(s) or as sub-module(s) of other module(s).

[0062] Program module(s), applications, or the like disclosed herein may include one or more software components including, for example, software objects, methods, data structures, or the like. Each such software component may include computer-executable instructions that, responsive to execution, cause at least a portion of the functionality described herein (e.g., one or more operations of the illustrative methods described herein) to be performed.

[0063] A software component may be coded in any of a variety of programming languages. An illustrative programming language may be a lower-level programming language such as an assembly language associated with a particular hardware architecture and/or operating system platform. A software component comprising assembly language instructions may require conversion into executable machine code by an assembler prior to execution by the hardware architecture and/or platform.

[0064] Another example programming language may be a higher-level programming language that may be portable across multiple architectures. A software component comprising higher-level programming language instructions may require conversion to an intermediate representation by an interpreter or a compiler prior to execution.

[0065] Other examples of programming languages include, but are not limited to, a macro language, a shell or command language, a job control language, a script language, a database query or search language, or a report writing language. In one or more example embodiments, a software component comprising instructions in one of the foregoing examples of programming languages may be executed directly by an operating system or other software component without having to be first transformed into another form.

[0066] A software component may be stored as a file or other data storage construct. Software components of a similar type or functionally related may be stored together such as, for example, in a particular directory, folder, or library. Software components may be static (e.g., pre-established or fixed) or dynamic (e.g., created or modified at the time of execution).

[0067] Software components may invoke or be invoked by other software components through any of a wide variety of mechanisms. Invoked or invoking software components may comprise other custom-developed application software, operating system functionality (e.g., device drivers, data storage (e.g., file management) routines, other common routines and services, etc.), or third-party software components (e.g., middleware, encryption, or other security software, database management software, file transfer or other network communication

software, mathematical or statistical software, image processing software, and format translation software).

[0068] Software components associated with a particular solution or system may reside and be executed on a single platform or may be distributed across multiple platforms. The multiple platforms may be associated with more than one hardware vendor, underlying chip technology, or operating system. Furthermore, software components associated with a particular solution or system may be initially written in one or more programming languages, but may invoke software components written in another programming language.

[0069] Computer-executable program instructions may be loaded onto a special-purpose computer or other particular machine, a processor, or other programmable data processing apparatus to produce a particular machine, such that execution of the instructions on the computer, processor, or other programmable data processing apparatus causes one or more functions or operations specified in the flow diagrams to be performed. These computer program instructions may also be stored in a computer-readable storage medium (CRSM) that upon execution may direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable storage medium produce an article of manufacture including instruction means that implement one or more functions or operations specified in the flow diagrams. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational elements or steps to be performed on the computer or other programmable apparatus to produce a computer-implemented process.

[0070] Additional types of CRSM that may be present in any of the devices described herein may include, but are not limited to, programmable random access memory (PRAM), SRAM, DRAM, RAM, ROM, electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technology, compact disc read-only memory (CD-ROM), digital versatile disc (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used

to store the information and which can be accessed. Combinations of any of the above are also included within the scope of CRSM. Alternatively, computer-readable communication media (CRCM) may include computer-readable instructions, program module(s), or other data transmitted within a data signal, such as a carrier wave, or other transmission. However, as used herein, CRSM does not include CRCM.

[0071] Although embodiments have been described in language specific to structural features and/or methodological acts, it is to be understood that the disclosure is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as illustrative forms of implementing the embodiments. Conditional language, such as, among others, "can," "could," "might," or "may," unless specifically stated otherwise, or otherwise understood within the context as used, is generally intended to convey that certain embodiments could include, while other embodiments do not include, certain features, elements, and/or steps. Thus, such conditional language is not generally intended to imply that features, elements, and/or steps are in any way required for one or more embodiments or that one or more embodiments necessarily include logic for deciding, with or without user input or prompting, whether these features, elements, and/or steps are included or are to be performed in any particular embodiment.

We Claim:

1. A system for managing network security of a computer network, the system comprising:

a processor:

a security module coupled to the processor, wherein the security module is configured to generate, by using a machine learning model, a risk score for each network transaction occurring in the computer network, wherein the risk score is indicative of a likelihood of a cyber attack; and

a response module coupled to the processor, wherein the response module is configured to initiate a plurality of responses based on the risk score for each of the network transactions, wherein the plurality of responses comprises:

restricting communication of a device associated with the network transaction when the risk score exceeds a predetermined threshold; and

recording the network transaction and notifying a user device, when the risk score is below the predetermined threshold and above a security notification threshold.

2. The system as claimed in claim 1, wherein the system further comprises a data ingestion module coupled to the processor and a training module coupled to the processor, wherein:

the data ingestion module is configured to provide network data to the ML model, wherein the network data comprises: network logs, firewall data, data from one or more intrusion detection systems; and

the training module is configured to train the ML model based on the network data.

3. The system as claimed in claim 2, wherein:

the data ingestion module is configured to receive cyber threat data from an external entity; and

the training module is configured to update the ML model based on

the cyber threat data.

4. The system as claimed in claim 1, wherein the system further comprises a feedback module coupled to the processor and a training module coupled to the processor, wherein:

the feedback module is configured to gather feedback data indicative of performance of the system; and

the training module is configured to train the ML model based on the feedback data.

5. The system as claimed in claim 1, wherein the security module is configured to generate a security report, wherein the security report comprises information about security risks, number of security attacks, and corresponding measures implemented.

6. A method for managing network security of a computer network, the method comprising:

generating a risk score for each network transaction occurring in the computer network using a machine learning model, wherein the risk score indicates the likelihood of a cyber attack; and

initiating a plurality of responses based on the risk score for each network transaction, wherein the responses include:

restricting communication of a device associated with the network transaction when the risk score exceeds a predetermined threshold.

recording the network transaction and notifying a user device when the risk score is below the predetermined threshold and above a security notification threshold.

7. The method as claimed in claim 1, wherein the method further comprises:

providing network data, including network logs, firewall data, and data from one or more intrusion detection systems, to the machine learning model; and

training the machine learning model based on the network data.

8. The method as claimed in claim 2, wherein the method further comprises:

receiving cyber threat data from an external entity; and

updating the machine learning model based on the cyber threat data.

9. The method as claimed in claim 1, wherein the method further comprises:

gathering feedback data indicative of the performance of the system; and

training the machine learning model based on the feedback data.

10. The method as claimed in claim 1, wherein the method further comprises:

generating a security report that includes information about security risks, the number of security attacks, and corresponding measures implemented.

For Chitkara University and
Bluest Mettle Solutions Private Limited

Tarun Khurana
Regd. Patent Agent [IN/PA-1325]
Dated: 11th July, 2023

ABSTRACT

CYBER ATTACK PREDICTION

The present invention discloses a system and method for managing network security in a computer network. The system consists of a processor, a security module, and a response module. Utilizing a machine learning model, the security module generates risk scores for network transactions, indicating the likelihood of cyber attacks. The response module triggers responses based on these risk scores, including communication restrictions for transactions exceeding a predetermined threshold and recording transactions while notifying user devices for risk scores falling within specified thresholds.

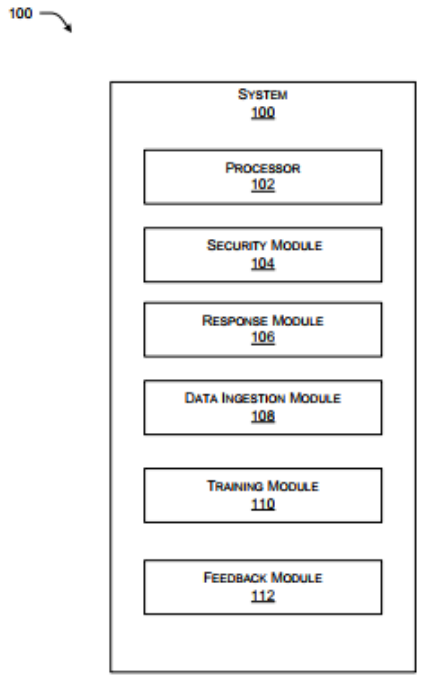

FIG. 1

For Chitkara University and
Bluest Mettle Solutions Private Limited

Tarun Khurana
Regd. Patent Agent [IN/PA-1325]
Dated: 11th July, 2023