United States Patent Application Publication	20250267164
Kind Code	A1
Publication Date	August 21, 2025
Inventor(s)	SAND; Ankur et al.

SYSTEMS AND METHODS FOR PROACTIVELY MONITORING THE INHERENT CYBER-TECH RISK OF SOFTWARE AND HARDWARE COMPONENTS

Abstract

Systems and methods for proactively monitoring the inherent cyber-tech risk of software and hardware components are disclosed. In one embodiment, a method for proactively monitoring a cyber risk of a computer program may include: (1) receiving, by a product/version risk assessment computer program executed by an electronic device and from a user computer program executed by a use electronic device, an identification of a plurality of proposed components to include in the computer program; (2) retrieving, by the product/version risk assessment computer program, vulnerability information for each of the plurality of proposed components, wherein the vulnerability information identifies a security vulnerability for the proposed component; (3) generating, by a product/version risk scoring computer program, a risk score for the computer program under development based on the vulnerability information; and (4) returning, by the vulnerability assessment computer program, the risk score to the user computer program.

Inventors: **SAND; Ankur (Cambridge, GB), WILSON; Ken (Millington, NJ), GRANT; Marty (Middletown, DE), WIJAYA; Herman (New York, NY), EDWARDS; David R. (Yokosuka, JP)**

Applicant: **JPMORGAN CHASE BANK, N.A.** (New York, NY)

Family ID: **1000008586911**

Appl. No.: **19/184680**

Filed: **April 21, 2025**

Related U.S. Application Data

parent US continuation 18061242 20221202 parent-grant-document US 12284201 child US 19184680

Publication Classification

Int. Cl.: **H04L9/40** (20220101)

U.S. Cl.:

CPC **H04L63/1433** (20130101);

Background/Summary

BACKGROUND OF THE INVENTION

1. Field Of The Invention

[0001] Embodiments generally relate to systems and methods for proactively monitoring the inherent cyber-tech risk of software and hardware components.

2. Description of the Related Art

[0002] Large organizations use many computer products. There are no proactive ways to identify and/or monitor the inherent cyber-tech risks of existing products before they are used. Due to a lack of product-based risk ranking methodology, currently cyber-risk considerations often start late in the Software Development Life Cycle (SDLC) which brings resource overhead and often results in unwanted cyber-risk exposures.

SUMMARY OF THE INVENTION

[0003] Systems and methods for proactively monitoring the inherent cyber-tech risk of software and hardware components are disclosed. In one embodiment, a method for proactively monitoring a cyber risk of a computer program may include: (1) receiving, by a product/version risk assessment computer program executed by an electronic device and from a user computer program executed by a use electronic device, an identification of a plurality of proposed components to include in the computer program; (2) retrieving, by the product/version risk assessment computer program, vulnerability information for each of the plurality of proposed components, wherein the vulnerability information identifies a security vulnerability for the proposed component; (3) generating, by a product/version risk scoring computer program, a risk score for the computer program under development based on the vulnerability information; and (4) returning, by the vulnerability assessment computer program, the risk score to the user computer program.

[0004] In one embodiment, the proposed components may include hardware components and/or software components.

[0005] In one embodiment, the vulnerability information is retrieved from an external threat and vulnerability tool/database.

[0006] In one embodiment, the vulnerability information is further retrieved from an internal threat and vulnerability tool/database.

[0007] In one embodiment, the risk score is based on a stack score for the computer program under development, a vulnerability density for each of the plurality of components, threat intelligence inputs for each of the plurality of components, patching maturity for each of the plurality of components, a lifecycle state for each of the plurality of components, and support coverage for each of the plurality of components.

[0008] In one embodiment, the risk score comprises a dynamic qualitative severity rating.

[0009] In one embodiment, the method may also include identifying, by the product/version risk assessment computer program, an alternate component for a potential component in response to the potential component having certain cyber-tech risk related information. In one embodiment, the alternate component may be identified using a trained machine learning engine.

[0010] According to another embodiment, a system may include a user electronic device executing a user computer program; an electronic device executing a product/version risk assessment computer program and a product/version risk scoring computer program; an internal threat and vulnerability tool/database; and an external threat and vulnerability tool/database. The user computer program receives an identification of a plurality of proposed components to include in a computer program, the product/version risk assessment computer program receives the identification of a plurality of proposed components and retrieves vulnerability information for each of the plurality of proposed components from the internal threat and vulnerability tool/database and the external threat and vulnerability tool/database; the product/version risk scoring computer program generates a risk score for the computer program under development based on the vulnerability information; and the vulnerability assessment computer program returns the risk score to the user computer program.
[0011] In one embodiment, the proposed components may include hardware components and/or software components.
[0012] In one embodiment, the risk score may be based on a stack score for the computer program under development, a vulnerability density for each of the plurality of components, threat intelligence inputs for each of the plurality of components, patching maturity for each of the plurality of components, a lifecycle state for each of the plurality of components, and support coverage for each of the plurality of components.
[0013] In one embodiment, the risk score may include a dynamic qualitative severity rating.
[0014] In one embodiment, the product/version risk assessment computer program identifies an alternate component for a potential component in response to the potential component having certain cyber-tech risk related information.
[0015] In one embodiment, the system may also include a trained machine learning engine, and the product/version risk assessment computer program identifies the alternate component using the trained machine learning engine.
[0016] According to another embodiment, a non-transitory computer readable storage medium, may include instructions stored thereon, which when read and executed by one or more computer processors, cause the one or more computer processors to perform steps comprising: receiving, from a user computer program executed by a user electronic device, an identification of a plurality of proposed components to include in a computer program; retrieving vulnerability information for each of the plurality of proposed components, wherein the vulnerability information identifies a security vulnerability for the proposed component; generating a risk score for the computer program under development based on the vulnerability information; and returning the risk score to the user computer program.
[0017] In one embodiment, the proposed components may include hardware components and/or software components.
[0018] In one embodiment, the vulnerability information is retrieved from an external threat and vulnerability tool/database.
[0019] In one embodiment, the vulnerability information is further retrieved from an internal threat and vulnerability tool/database.
[0020] In one embodiment, the risk score may be based on a stack score for the computer program under development, a vulnerability density for each of the plurality of components, threat intelligence inputs for each of the plurality of components, patching maturity for each of the plurality of components, a lifecycle state for each of the plurality of components, and support coverage for each of the plurality of components.
[0021] In one embodiment, the risk score may include a dynamic qualitative severity rating.
[0022] In one embodiment, the non-transitory computer readable storage medium of claim **15**, may also include instructions stored thereon, which when read and executed by one or more computer processors, cause the one or more computer processors to perform identify an alternate component

for a potential component in response to the potential component having certain cyber-tech risk related information using a trained machine learning engine.

Description

BRIEF DESCRIPTION OF THE DRAWINGS
[0023] In order to facilitate a fuller understanding of the present invention, reference is now made to the attached drawings. The drawings should not be construed as limiting the present invention but are intended only to illustrate different aspects and embodiments.
[0024] FIG. **1** depicts a system for proactively monitoring the inherent cyber-tech risk of software and hardware components according to an embodiment.
[0025] FIG. **2** depicts a method for proactively monitoring the inherent cyber-tech risk of software and hardware components according to one embodiment.
[0026] FIG. **3** depicts an exemplary computing system for implementing aspects of the present disclosure.
DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS
[0027] Embodiments are directed to systems and methods for proactively monitoring the inherent cyber-tech risk of software and hardware components.
[0028] Embodiments may provide a proactive, risk-driven approach to calculate a transparent related risk status for a computer program/components before and after it is developed.
[0029] Embodiments may include several components: risk scoring and visualization, recommendations, forecast/prediction, and subscription/alerting.
[0030] Embodiments may provide at least some of the following benefits: a better developer experience by pre-informing developers of vulnerabilities early in development and for current/future usages; dynamic cyber-tech risk representation via, for example, quantitative and qualitative methods, to support situational changes; increase time to market by identifying vulnerabilities early in development; reduction in costs due to less remediation; improved data driven transparency and insights by integral artificial intelligence and machine learning components; universal coverage (e.g., hardware, software, firmware, Internet of Things (IoT) appliances, etc.), dashboard based visualization schemes, etc.
[0031] Referring to FIG. **1**, a system for proactively monitoring the inherent cyber-tech risk of software and hardware components is disclosed according to an embodiment. System **100** may include electronic device **110**, which may be any suitable electronic device, such as servers (e.g., cloud based and/or physical), computers (e.g., workstation, desktop, notebook, laptop, tablet, etc.), smart devices (e.g., smartphones, smartwatches, etc.) Internet of Things (IoT) appliances, etc. Electronic device **110** may execute one or more computer programs or applications, including, for example, product/version risk assessment computer program **112**.
[0032] Product/version risk assessment computer program **112** may be a distributed computer application that may also be executed by user electronic device **120**. In another embodiment, user electronic device **120** may access product/version risk assessment computer program **112** via a user interface (not shown), such as a browser.
[0033] Product/version risk assessment computer program **112** may interface with product threat and vulnerability tools/databases, including external threat and vulnerability tool(s)/database(s) **130** and internal threat and vulnerability tool(s)/database(s) **135**. External threat and vulnerability tool(s)/database(s) **130** may identify vulnerabilities for hardware and software components, such as security-related software flaws, misconfigurations, impact metrics, etc. External threat and vulnerability tool(s)/database(s) **130** may be provided by third parties. Examples of external threat and vulnerability tool(s)/database(s) **130** include the National Vulnerability Database, Technopedia, vendor specific security release bulletins, etc.

[0034] Internal threat and vulnerability tool(s)/database(s) **135** may include vulnerability information, policies, and tools for the organization. In one embodiment, internal threat and vulnerability tool(s)/database(s) **135** may impose additional restrictions on any software components.
[0035] User electronic device **120** may be any suitable electronic device, including computers (e.g., workstation, desktop, notebook, laptop, tablet, etc.), smart devices (e.g., smartphones, smartwatches, etc.) Internet of Things (IoT) appliances, etc. Using user application **125**, a user, such as a developer, may identify proposed software components to be used in a computer program being developed. The user may also identify potential hardware components that may execute the computer program. Product/version risk assessment computer program **112** may receive an identification of the software and/or hardware components, and may retrieve vulnerability information from threat and vulnerability tool(s)/database(s) **130** and/or internal threat and vulnerability tool(s)/database(s) **135**, and may assess the computer program based on the vulnerabilities.
[0036] In one embodiment, product/version risk assessment computer program **112** may generate a risk score for the program using product/version risk scoring computer program **114**. Product/version risk scoring computer program **114** may consider factors including, for example, a stack score (e.g., hardware, firmware/hypervisors, operating system, application/middleware, and data for the computer program may each have a different risk/priority level), a vulnerability density (e.g., the number of vulnerabilities identified within the components), threat intelligence inputs (e.g., any Common Vulnerabilities and Exposures (CVE) that is associated, by various Threat Intelligence sources, with threat actors/advanced persistent threats, public exploit code presence, associated with supply chain compromises, association with malware, etc.), patching maturity (e.g., how often vendors for the components release patches for vulnerabilities, organizations specific patching practices levels, etc.), controlled utilization (e.g., nature of the component usage with regard to ownership), lifecycle state for the component (e.g., not approved/prohibited, research, endorsed for use, migrate to new version or different product, etc.), and support coverage for the program (e.g., no support, limited support, full support), etc. Each of the factors may be given a different weight (e.g., the stack score for a data component may be higher than that for its hardware, a product with high vulnerability density carries a different weight from a product with low vulnerability density, and a product with existing critical or severe vulnerabilities as rated by the organization may carry extra weight in the overall product risk score calculation).
[0037] In one embodiment, the weightings may be assigned manually or may be assigned using a machine learning engine, such as machine learning engine **116**.
[0038] In one embodiment, product/version risk scoring computer program **114** may take ordinal data, such as vulnerability density, threat intelligence inputs, lifecycle and support status of the affected product/version, etc., and may assign relative importance rankings to the ordinal data as ratio values.
[0039] In one embodiment, the risk scores may be mapped to qualitative severity ratings (e.g., critical, severe, high, moderate, and low). For example, a risk score of 4.0 may have an associated severity rating of moderate. This mapping may be customized based on the organization's needs.
[0040] Using machine learning engine **116**, product/version risk assessment computer program **112** may also identify alternate components that may lower the risk assessment.
[0041] In one embodiment, once development of the computer program is complete, the computer program may be deployed to production environment **140**.
[0042] In one embodiment, product/version risk assessment computer program **112** may also assess computer programs under development in development environment **150**, such as new versions of the computer program that are under development.
[0043] Once the computer program is deployed, product/version risk assessment computer program **112** may periodically re-assess the risk for the computer program and may provide alerts and

recommendations as warranted.
[0044] In one embodiment, trained machine learning engine **116** may predict when a computer program may have a risk assessment that may be above a threshold set against the operating environment.
[0045] Referring to FIG. **2**, a method for proactively monitoring the inherent cyber-tech risk of software and hardware components is disclosed according to an embodiment.
[0046] In step **205**, a user, such as a developer, of a computer program may select components that are proposed to include in a computer program under development. The components may include commercial software components, open-source software components, internal software components, hardware, etc.
[0047] The user may identify the components in a user interface of a product/version risk assessment computer program. For example, a library of available components may be provided, and the user may identify the proposed components from the library.
[0048] In step **210**, a product/version risk assessment computer program may access a product/version risk scoring computer program. In one embodiment, the product/version risk scoring computer program may receive the selection of components and may access external and/or internal threat and vulnerability tool(s)/database(s) to identify vulnerabilities in the components and cyber threats. For example, the product/version risk scoring computer program may access external product threat and vulnerability tool(s)/database(s), such as the National Vulnerability Database, Technopedia, etc., to retrieve vulnerabilities for the components. In addition, the product/version risk computer program may access internal product vulnerability databases, which may provide vulnerability information and assessments that may be specific to the organization.
[0049] In one embodiment, if there is a conflict between information in the external product vulnerability database and the internal product vulnerability database, the information in the internal product vulnerability may control.
[0050] In addition, the product/version risk scoring computer program may also identify factors associated with cyber risk to the program, such as a stack score, a vulnerability density, threat intelligence inputs, patching maturity, and controlled utilization. It may also identify factors associated with technical risk to the program, such as the lifecycle state of the program and support coverage for the program.
[0051] In step **215**, the product/version risk scoring computer program may score the cyber-tech risk for the computer program. In one embodiment, the risk assessment or scoring may consider various risk elements of context and material consequences of the associated threats and vulnerabilities, etc. against the given products or versions during the risk ranking. It may take ordinal data, such as vulnerability density, threat intelligence inputs, lifecycle and support status of the affected product/version, etc., and may assign relative importance rankings as ratio values. The risk scores may be mapped to qualitative severity ratings (e.g., critical, severe, high, moderate, and low).
[0052] In one embodiment, the evaluation criteria, including vulnerabilities and other associated metrics, may be assigned different weights.
[0053] In step **220**, the product/version risk assessment computer program may return the risk for the identified components. For example, the product/version risk assessment computer program may provide a qualitative severity rating or a risk score to the user. The risk assessment or risk score may be provided as high-medium-low, red-amber-green, a numeric score, etc. The risk assessment or risk score may be provided for the entire computer program, or for individual components or groups of components.
[0054] In step **225**, the product/version risk assessment computer program may provide recommendations for alternate components that have less risk. For example, if a certain component is contributing to the risk assessment or risk score, the product/version risk assessment computer program may identify an alternate component for the certain component.

[0055] In one embodiment, the alternate component may be identified from an internal or external threat and vulnerability tool(s)/database(s).
[0056] In one embodiment, a trained machine learning engine may identify and provide recommendations on alternate products. For example, the trained machine learning engine may be trained on historical data to identify alternate components that have similar functionality and acceptable risk score that have been deployed to other computer programs.
[0057] The trained machine learning engine may also predict whether the alternate component can be used (e.g., no dependency issues, no interaction issues with other components, etc.).
[0058] In one embodiment, the trained machine learning engine may learn the user's behavior/search data and may recommend alternative components from the same product domain or type, which have lower risk ratings, as alternatives to the components that the user is searching for and available within the organizations inventory system.
[0059] In one embodiment, the trained machine learning engine may predict/forecast the product's risk ratings for the next N-months based on the historical risk ratings of the product and/or versions and the ongoing threat intelligence inputs from various sources, affected vendors patch release frequency/trends, etc.
[0060] In one embodiment, the trained machine learning engine may learn the user's search and subscription data, and may notify the user of any identified vulnerabilities and/or change in risk ratings within the products/versions which the user uses or has subscribed to. Embodiments may also monitor security hygiene best practices, such as notifying the stakeholders if their affected versions are behind pre-configured latest version, and may also reassess the organization's risk profiles and risk appetites values derived from the various available products regularly, based on feedback from stakeholders throughout the organization.
[0061] In step **230**, the user may complete the computer program using the components or alternate components. When the computer program is complete, it may be deployed to a production environment for use.
[0062] In step **235**, the product/version risk assessment computer program may reassess the components in completed program. For example, the reassessment may be performed periodically, on-demand, or as otherwise necessary and/or desired. In one embodiment, the product/version risk assessment computer program may assess the vulnerability in a manner similar to that described in steps **210**, **215**, and **220**, above.
[0063] In step **235**, if there is a new vulnerability, or the risk assessment is above a certain level, in step **240**, the product/version risk assessment computer program may provide an alert to the user. The product/version risk assessment computer program may also provide recommendations on alternate components for any vulnerable components.
[0064] In one embodiment, if the risk assessment is at or above a certain level, the product/version risk assessment computer program may disable any affected computer program/components within the inventory system until the vulnerability is addressed.
[0065] FIG. **3** depicts an exemplary computing system for implementing aspects of the present disclosure. FIG. **3** depicts exemplary computing device **300**. Computing device **300** may represent the system components described herein. Computing device **300** may include processor **305** that may be coupled to memory **310**. Memory **310** may include volatile memory. Processor **305** may execute computer-executable program code stored in memory **310**, such as software programs **315**. Software programs **315** may include one or more of the logical steps disclosed herein as a programmatic instruction, which may be executed by processor **305**. Memory **310** may also include data repository **320**, which may be nonvolatile memory for data persistence. Processor **305** and memory **310** may be coupled by bus **330**. Bus **330** may also be coupled to one or more network interface connectors **340**, such as wired network interface **342** or wireless network interface **344**. Computing device **300** may also have user interface components, such as a screen for displaying graphical user interfaces and receiving input from the user, a mouse, a keyboard and/or other

input/output components (not shown).
[0066] The disclosures of U.S. Provisional Patent Application Ser. Nos. 63/126,935 and 63/138,951, U.S. patent application Ser. No. 17/538,763, and U.S. patent application Ser. No. 17/664,579 are hereby incorporated, by reference, in their entireties.
[0067] Although several embodiments have been disclosed, it should be recognized that these embodiments are not exclusive to each other, and features from one embodiment may be used with others.
[0068] Hereinafter, general aspects of implementation of the systems and methods of embodiments will be described.
[0069] Embodiments of the system or portions of the system may be in the form of a "processing machine," such as a general-purpose computer, for example. As used herein, the term "processing machine" is to be understood to include at least one processor that uses at least one memory. The at least one memory stores a set of instructions. The instructions may be either permanently or temporarily stored in the memory or memories of the processing machine. The processor executes the instructions that are stored in the memory or memories in order to process data. The set of instructions may include various instructions that perform a particular task or tasks, such as those tasks described above. Such a set of instructions for performing a particular task may be characterized as a program, software program, or simply software.
[0070] In one embodiment, the processing machine may be a specialized processor.
[0071] In one embodiment, the processing machine may be a cloud-based processing machine, a physical processing machine, or combinations thereof.
[0072] As noted above, the processing machine executes the instructions that are stored in the memory or memories to process data. This processing of data may be in response to commands by a user or users of the processing machine, in response to previous processing, in response to a request by another processing machine and/or any other input, for example.
[0073] As noted above, the processing machine used to implement embodiments may be a general-purpose computer. However, the processing machine described above may also utilize any of a wide variety of other technologies including a special purpose computer, a computer system including, for example, a microcomputer, mini-computer or mainframe, a programmed microprocessor, a micro-controller, a peripheral integrated circuit element, a CSIC (Customer Specific Integrated Circuit) or ASIC (Application Specific Integrated Circuit) or other integrated circuit, a logic circuit, a digital signal processor, a programmable logic device such as a FPGA (Field-Programmable Gate Array), PLD (Programmable Logic Device), PLA (Programmable Logic Array), or PAL (Programmable Array Logic), or any other device or arrangement of devices that is capable of implementing the steps of the processes disclosed herein.
[0074] The processing machine used to implement embodiments may utilize a suitable operating system.
[0075] It is appreciated that in order to practice the method of the embodiments as described above, it is not necessary that the processors and/or the memories of the processing machine be physically located in the same geographical place. That is, each of the processors and the memories used by the processing machine may be located in geographically distinct locations and connected so as to communicate in any suitable manner. Additionally, it is appreciated that each of the processor and/or the memory may be composed of different physical pieces of equipment. Accordingly, it is not necessary that the processor be one single piece of equipment in one location and that the memory be another single piece of equipment in another location. That is, it is contemplated that the processor may be two pieces of equipment in two different physical locations. The two distinct pieces of equipment may be connected in any suitable manner. Additionally, the memory may include two or more portions of memory in two or more physical locations.
[0076] To explain further, processing, as described above, is performed by various components and various memories. However, it is appreciated that the processing performed by two distinct

components as described above, in accordance with a further embodiment, may be performed by a single component. Further, the processing performed by one distinct component as described above may be performed by two distinct components.

[0077] In a similar manner, the memory storage performed by two distinct memory portions as described above, in accordance with a further embodiment, may be performed by a single memory portion. Further, the memory storage performed by one distinct memory portion as described above may be performed by two memory portions.

[0078] Further, various technologies may be used to provide communication between the various processors and/or memories, as well as to allow the processors and/or the memories to communicate with any other entity; i.e., so as to obtain further instructions or to access and use remote memory stores, for example. Such technologies used to provide such communication might include a network, the Internet, Intranet, Extranet, a LAN, an Ethernet, wireless communication via cell tower or satellite, or any client server system that provides communication, for example. Such communications technologies may use any suitable protocol such as TCP/IP, UDP, or OSI, for example.

[0079] As described above, a set of instructions may be used in the processing of embodiments. The set of instructions may be in the form of a program or software. The software may be in the form of system software or application software, for example. The software might also be in the form of a collection of separate programs, a program module within a larger program, or a portion of a program module, for example. The software used might also include modular programming in the form of object-oriented programming. The software tells the processing machine what to do with the data being processed.

[0080] Further, it is appreciated that the instructions or set of instructions used in the implementation and operation of embodiments may be in a suitable form such that the processing machine may read the instructions. For example, the instructions that form a program may be in the form of a suitable programming language, which is converted to machine language or object code to allow the processor or processors to read the instructions. That is, written lines of programming code or source code, in a particular programming language, are converted to machine language using a compiler, assembler or interpreter. The machine language is binary coded machine instructions that are specific to a particular type of processing machine, i.e., to a particular type of computer, for example. The computer understands the machine language.

[0081] Any suitable programming language may be used in accordance with the various embodiments. Also, the instructions and/or data used in the practice of embodiments may utilize any compression or encryption technique or algorithm, as may be desired. An encryption module might be used to encrypt data. Further, files or other data may be decrypted using a suitable decryption module, for example.

[0082] As described above, the embodiments may illustratively be embodied in the form of a processing machine, including a computer or computer system, for example, that includes at least one memory. It is to be appreciated that the set of instructions, i.e., the software for example, that enables the computer operating system to perform the operations described above may be contained on any of a wide variety of media or medium, as desired. Further, the data that is processed by the set of instructions might also be contained on any of a wide variety of media or medium. That is, the particular medium, i.e., the memory in the processing machine, utilized to hold the set of instructions and/or the data used in embodiments may take on any of a variety of physical forms or transmissions, for example. Illustratively, the medium may be in the form of a compact disc, a DVD, an integrated circuit, a hard disk, a floppy disk, an optical disc, a magnetic tape, a RAM, a ROM, a PROM, an EPROM, a wire, a cable, a fiber, a communications channel, a satellite transmission, a memory card, a SIM card, or other remote transmission, as well as any other medium or source of data that may be read by the processors.

[0083] Further, the memory or memories used in the processing machine that implements

embodiments may be in any of a wide variety of forms to allow the memory to hold instructions, data, or other information, as is desired. Thus, the memory might be in the form of a database to hold data. The database might use any desired arrangement of files such as a flat file arrangement or a relational database arrangement, for example.

[0084] In the systems and methods, a variety of “user interfaces” may be utilized to allow a user to interface with the processing machine or machines that are used to implement embodiments. As used herein, a user interface includes any hardware, software, or combination of hardware and software used by the processing machine that allows a user to interact with the processing machine. A user interface may be in the form of a dialogue screen for example. A user interface may also include any of a mouse, touch screen, keyboard, keypad, voice reader, voice recognizer, dialogue screen, menu box, list, checkbox, toggle switch, a pushbutton or any other device that allows a user to receive information regarding the operation of the processing machine as it processes a set of instructions and/or provides the processing machine with information. Accordingly, the user interface is any device that provides communication between a user and a processing machine. The information provided by the user to the processing machine through the user interface may be in the form of a command, a selection of data, or some other input, for example.

[0085] As discussed above, a user interface is utilized by the processing machine that performs a set of instructions such that the processing machine processes data for a user. The user interface is typically used by the processing machine for interacting with a user either to convey information or receive information from the user. However, it should be appreciated that in accordance with some embodiments of the system and method, it is not necessary that a human user actually interact with a user interface used by the processing machine. Rather, it is also contemplated that the user interface might interact, i.e., convey and receive information, with another processing machine, rather than a human user. Accordingly, the other processing machine might be characterized as a user. Further, it is contemplated that a user interface utilized in the system and method may interact partially with another processing machine or processing machines, while also interacting partially with a human user.

[0086] It will be readily understood by those persons skilled in the art that embodiments are susceptible to broad utility and application. Many embodiments and adaptations of the present invention other than those herein described, as well as many variations, modifications and equivalent arrangements, will be apparent from or reasonably suggested by the foregoing description thereof, without departing from the substance or scope.

[0087] Accordingly, while the embodiments of the present invention have been described here in detail in relation to its exemplary embodiments, it is to be understood that this disclosure is only illustrative and exemplary of the present invention and is made to provide an enabling disclosure of the invention. Accordingly, the foregoing disclosure is not intended to be construed or to limit the present invention or otherwise to exclude any other such embodiments, adaptations, variations, modifications or equivalent arrangements.

Claims

1. A method, comprising: receiving, by a product/version risk assessment computer program executed by an electronic device and from a user computer program executed by a user electronic device, an identification of a plurality of proposed components to include in the computer program under development; retrieving, by the product/version risk assessment computer program, vulnerability information for each of the plurality of proposed components, wherein the vulnerability information identifies a security vulnerability for the proposed component; generating, by a product/version risk scoring computer program, a risk score for the computer program under development based on the vulnerability information; and returning, by the product/version risk assessment computer program, the risk score to the user computer program.

2. The method of claim 1, wherein the plurality of proposed components comprise hardware components and/or software components.
3. The method of claim 1, wherein the vulnerability information is retrieved from an external threat and vulnerability tool/database.
4. The method of claim 3, wherein the vulnerability information is further retrieved from an internal threat and vulnerability tool/database.
5. (canceled)
6. The method of claim 1, wherein the risk score comprises a dynamic qualitative severity rating.
7. The method of claim 1, further comprising: identifying, by the product/version risk assessment computer program, an alternate component for one of the plurality of proposed components in response to the proposed component having certain cyber-tech risk related information.
8. The method of claim 7, wherein the alternate component is identified using a trained machine learning engine.
9. A system, comprising: a user electronic device executing a user computer program; an electronic device executing a product/version risk assessment computer program and a product/version risk scoring computer program; an internal threat and vulnerability tool/database; and an external threat and vulnerability tool/database; wherein: the user computer program receives an identification of a plurality of proposed components to include in a computer program under development; the product/version risk assessment computer program receives the identification of a plurality of proposed components and retrieves vulnerability information for each of the plurality of proposed components from the internal threat and vulnerability tool/database and the external threat and vulnerability tool/database; the product/version risk scoring computer program generates a risk score for the computer program under development based on the vulnerability information; and the product/version risk assessment computer program returns the risk score to the user computer program.
10. The system of claim 9, wherein the plurality of proposed components comprise hardware components and/or software components.
11. (canceled)
12. The system of claim 9, wherein the risk score comprises a dynamic qualitative severity rating.
13. The system of claim 9, wherein the product/version risk assessment computer program identifies an alternate component for one of the plurality of proposed components in response to the proposed component having certain cyber-tech risk related information.
14. The system of claim 13, further comprising a trained machine learning engine, wherein the product/version risk assessment computer program identifies the alternate component using the trained machine learning engine.
15. A non-transitory computer readable storage medium, including instructions stored thereon, which when read and executed by one or more computer processors, cause the one or more computer processors to perform steps comprising: receiving, from a user computer program executed by a use electronic device, an identification of a plurality of proposed components to include in a computer program under development; retrieving vulnerability information for each of the plurality of proposed components, wherein the vulnerability information identifies a security vulnerability for the proposed component; generating a risk score for the computer program under development based on the vulnerability information; and returning the risk score to the user computer program.
16. The non-transitory computer readable storage medium of claim 15, wherein the plurality of proposed components comprise hardware components and/or software components.
17. The non-transitory computer readable storage medium of claim 15, wherein the vulnerability information is retrieved from an external threat and vulnerability tool/database.
18. The non-transitory computer readable storage medium of claim 17, wherein the vulnerability information is further retrieved from an internal threat and vulnerability tool/database.

19. (canceled)

20. The non-transitory computer readable storage medium of claim 15, further including instructions stored thereon, which when read and executed by one or more computer processors, cause the one or more computer processors to perform identify an alternate component for one of the plurality of proposed components in response to the proposed component having certain cyber-tech risk related information using a trained machine learning engine.