

DEVICE-CODE-BASED RISK SCANNING METHOD, SYSTEM, AND COMPUTING DEVICE

TECHNICAL FIELD

[0001] The present invention relates to the field of code risk prevention and control, and in particular, to a device-code-based risk scanning method, system, and computing device.

BACKGROUND

[0002] When performing vulnerability detection on a website backend system on an online server (for example: a website server) and a hardware device operating online, the intermediate code and binary code on the server, as well as the binary code in the hardware device, are all human-unrecognizable code (for example: Linux Kernel, Web Server, etc.). Moreover, the server and hardware device are already operating online. Therefore, in the current existing technology, a network packet simulating an attack is sent to the server and the hardware device through a network according to the behavior description of a specific vulnerability. After the server and hardware device receive the network packet, whether the response of the server and the hardware device conforms to the behavior description of the specific vulnerability is monitored. When they conform, it indicates that the specific vulnerability exists in the server and the hardware device operating online, thereby providing vulnerability scanning functionality for operators of the server and the online hardware device.

[0003] When analyzing vulnerabilities and scanning risks of online servers and/or hardware devices, since the code running in the servers and/or hardware devices is mainly intermediate code or binary code, in the absence of source code, it is impossible to compare the source code of patches with the source code of the online system; source code comparison cannot be implemented without source code. However, vulnerability analysis using simulated packets requires the support of a network environment. In the case of offline status or abnormal network operation, the simulated packets cannot be sent to the online servers and/or hardware devices to be tested, and vulnerability analysis and risk monitoring using simulated packets also cannot be implemented. Furthermore, when the trigger of a specific vulnerability cannot be achieved through simulated packets, the specific vulnerability cannot be detected by means of simulated packets.

[0004] For IoT sensors, network communication devices (for example: WiFi wireless routers), or other embedded devices with limited computing capabilities, code component analysis cannot be directly performed in such devices for computing like antivirus scanning software on personal computers. On one hand, the design of hardware and software specifications of personal computers not only considers the computing power required by the operating system itself but also reserves sufficient computing power for applications installed by users. On the other hand, personal computers provide a general-purpose operating platform that is not designed for a single or specific function, allowing different applications to be expanded into the operating environment for use through installation; therefore, antivirus scanning software for personal computers can be installed on personal computers for use. For IoT sensors, network communication devices (for example: WiFi wireless routers), or other embedded devices with limited computing capabilities, on one hand, their function is not to serve as a general-purpose operating platform but to provide a single or specific service, and they do not have the condition to arbitrarily install applications to expand their working capabilities. On the other hand, the design of their hardware and software specifications mainly optimizes the single or specific service they provide; therefore, code component analysis is difficult to operate in such devices or apparatuses.

[0005] Based on the above description, in the technical field of risk scanning for information security, how to perform vulnerability scanning on hardware devices that lack source code, operate online, and/or have hardware and software specifications designed for a single or specific function is an important technical issue. Currently, there is no device code risk scanning method, software system, or computing device that can provide a vulnerability analysis risk scanning method and a system using the method for devices in the state of lacking source code of the device system, offline network, insufficient computing power, and/or inability to expand applications, so it is still urgently needed to be developed.

SUMMARY

[0006] To address the above technical issues, the present invention provides a device-code-based risk scanning method, system, and computing device. The present invention automatically sends scanning results to users through self-generated update records, improving the efficiency and accuracy of risk scanning.

[0007] Solution 1:

- [0008] A device-code-based risk scanning method, including:
- [0009] obtaining a device-code, and sending the device-code from a first device to a second device; where the device-code is acquired from the first device and includes at least one software code;
- [0010] generating an update record; where the update record includes a relationship record between the device-code and the first device and/or the second device; and
- [0011] sending, when a security risk is detected in the device-code, a scanning result to the first device and/or the second device according to the update record.
- [0012] Further, sending the scanning result refers to sending a notification of the scanning result to at least one of a user, an operator, or a manager of the first device and the second device via at least one network communication channel.
- [0013] Further, the at least one software code includes at least one binary code.
- [0014] Further, obtaining the device-code specifically includes:
- [0015] acquiring the device-code from a storage medium of the first device according to an update request.
- [0016] Further, the approach for obtaining the device-code further includes:
- [0017] obtaining, from the storage medium and when the update request is acquired, a device-code corresponding to version information or device specification information according to the version information or the device specification information in the update request as the device-code; where the device specification information includes at least one of a product specification, a hardware specification, and a firmware specification of a device.
- [0018] Further, sending the device-code to the second device specifically includes:
- [0019] sending the device-code to the second device according to network location information of the second device contained in the update request.
- [0020] Further, the security risk is detected in the device-code specifically includes:
- [0021] comparing, when at least one code analysis event is triggered, component contents of at least one risk code with the at least one software code; where the at least one risk code includes at least one of a patch code associated with the at least one software code, a malicious program code, and any computer program code written in a computer programming

language.

[0022] Further, the at least one risk code is pre-collected according to at least one risk code login list, or collected when a risk code is added to or updated in the at least one risk code login list.

[0023] Further, the at least one code analysis event is triggered when a composition list including the at least one software code is obtained, or triggered when the at least one risk code is added or updated.

[0024] Further, comparing the at least one risk code with the at least one software code specifically includes:

[0025] extracting at least one first component feature corresponding to the at least one risk code;

[0026] extracting at least one second component feature corresponding to the at least one software code; and

[0027] comparing the at least one first component feature with the at least one second component feature.

[0028] Further, the at least one first component feature is obtained by analyzing code composition content and/or a code execution result corresponding to the at least one risk code.

[0029] Further, the code composition content includes the at least one risk code and/or at least one first intermediate code obtained by compiling the at least one risk code, where the code execution result includes semantic information revealed during simulation execution of the at least one risk code and/or the at least one first intermediate code when performing simulation execution on the at least one risk code and/or the at least one first intermediate code.

[0030] Further, the at least one second component feature is obtained by analyzing at least one code composition content and/or at least one code execution result corresponding to the at least one software code.

[0031] Further, the at least one code composition content includes at least one binary code and/or at least one second intermediate code obtained by disassembling the at least one binary code, where the at least one code execution result includes semantic information revealed during simulation execution of the at least one binary code and/or the at least one second intermediate code when performing simulation execution on the at least one binary code and/or

the at least one second intermediate code.

[0032] Further, a programming language structure of the intermediate code is declared by an intermediate language, an intermediate representation, a structured language, and/or a structured protocol.

[0033] Further, before the code analysis event is triggered, the method further includes:

[0034] detecting at least one code analysis event according to a composition list of the at least one software code, where the code analysis event corresponding to the at least one risk code is one of the at least one code analysis event.

[0035] Further, the composition list is obtained by scanning and/or analyzing a component composition of the device-code when the device-code is acquired.

[0036] Further, the composition list includes metadata of the device-code itself and/or metadata of the at least one software code contained in the component composition of the device-code.

[0037] Further, the at least one software code is present in the component composition of the device-code in forms includes a system core, a driver program, an application program, a component, and/or a class library.

[0038] Further, the method includes:

[0039] monitoring a risk code login list corresponding to each software code contained in the composition list according to the composition list, so as to detect a code analysis event by monitoring the abnormal changes in the risk code login list.

[0040] Further, the approach for obtaining the composition list specifically includes:

[0041] scanning and/or analyzing, when the device-code is a system image file, a component composition contained in the system image file to identify the at least one software code present in the system image file, so as to obtain the composition list of the at least one software code; or

[0042] scanning and/or analyzing, when the device-code is a software code package, a component composition contained in the software code package to identify the at least one software code present in the software code package, so as to obtain the composition list of the at least one software code.

[0043] Further, sending the scanning result to the first device and/or the second device

according to the update record specifically includes:

[0044] searching the update record to obtain the relationship record in the update record;
and

[0045] sending the scanning result to at least one of a user, an operator, or a manager of the first device and the second device according to the relationship record.

[0046] Further, the scanning result includes at least one of a scanning report and scanning status notification information; where the scanning report is used to show a result of comparing component contents of the at least one software code with the at least one risk code; the scanning status notification information is used to provide a notification of a scanning status to at least one of a user, an operator, or a manager of the first device and the second device.

[0047] Solution 2:

[0048] A device-code-based risk scanning system, including: a code sending module, an update record generating module, and a scanning result notification module; where

[0049] the code sending module is configured to obtain a device-code, and send the device-code from a first device to a second device; where the device-code is acquired from the first device and includes at least one software code;

[0050] the update record generating module is configured to generate an update record; where the update record includes a relationship record between the device-code and the first device and/or the second device; and

[0051] the scanning result notification module is configured to send a scanning result to the first device and/or the second device according to the update record when a security risk is detected in the device-code.

[0052] Further, the system further includes: a first feature extraction unit, a second feature extraction unit, and a feature comparison unit; where

[0053] the first feature extraction unit is configured to extract at least one first component feature corresponding to at least one risk code;

[0054] the second feature extraction unit is configured to extract at least one second component feature corresponding to the at least one software code; and

[0055] the feature comparison unit is configured to compare the at least one first component feature with the at least one second component feature.

[0056] Further, the first feature extraction unit or the second feature extraction unit includes at least one analysis unit and at least one retrieval unit; where

[0057] the at least one analysis unit is configured to obtain the at least one first component feature by analyzing code composition content and/or a code execution result corresponding to the at least one risk code, and/or obtain the at least one second component feature by analyzing code composition content and/or a code execution result corresponding to the at least one software code; and

[0058] the at least one retrieval unit is configured to extract the at least one first component feature and/or the at least one second component feature obtained after analysis.

[0059] Further, the system further includes: at least one detection unit; where

[0060] the at least one detection unit is configured to monitor a risk code login list corresponding to each software code contained in a composition list according to the composition list of the at least one software code, so as to detect a code analysis event by monitoring abnormal changes in the risk code login list.

[0061] Solution 3:

[0062] A computing device for device-code-based risk scanning, including: a code sending module, an update record generating module, and a scanning result notification module; where

[0063] the code sending module is configured to obtain a device-code, and send the device-code from a first device to a second device; where the device-code is acquired from the first device and includes at least one software code;

[0064] the update record generating module is configured to generate an update record; where the update record includes a relationship record between the device-code and the first device and/or the second device; and

[0065] the scanning result notification module is configured to send a scanning result to the first device and/or the second device according to the update record when a security risk is detected in the device-code.

BRIEF DESCRIPTION OF THE DRAWINGS

[0066] FIG. 1 is a schematic flow diagram of a device-code-based risk scanning method

of the present invention.

[0067] FIG. 2 is a schematic structural diagram of a device-code-based risk scanning system of the present invention.

[0068] FIG. 3 is a schematic structural diagram of a device-code-based risk scanning computing device of the present invention.

[0069] Reference numerals:

[0070] 1. Risk scanning system; 2. Code sending module; 3. Update record generating module; 4. Scanning result notification module; 5. First device; 6. Second device; 11. Computing device; 21. Code sending module; 31. Update record generating module; 41. Scanning result notification module.

DETAILED DESCRIPTION

[0071] The present invention can be more fully understood through the following description, including the following glossary of terms and concluding examples. For the sake of brevity, publications cited in this specification, including the disclosure content of patents, are incorporated herein by reference.

[0072] The following provides detailed descriptions of embodiments. The embodiments of the present invention are non-limiting and merely represent various aspects and features of the present invention. The embodiments are only for illustrative purposes and do not limit the scope of protection intended by the present invention. In addition, unnecessary components or components that can be completed by conventional technologies are omitted from the drawings in the embodiments to clearly show the technical characteristics of the present invention.

[0073] In the most restrictive technical sense, the code composition content of the “software code” described herein includes any binary code for loading and execution, intermediate code, and/or any computer program code written in a computer programming language. The “device-code” described herein refers to the code that runs on a device and provides logical computing and services for the device. The device-code includes binary code, intermediate code, and/or any computer program code written in a computer programming language, that is, the device-code includes at least one software code. The at least one software code is present in the component composition of the device-code in the form of components including a system core, a driver program, an application program, a component, and/or a class

library. The “binary code” described herein may include any code in binary representation format. The “intermediate code” described herein may include any code represented by a structure declared by an intermediate language, an intermediate representation, a structured language, and/or a structured protocol; for example, Java Bytecode represented in accordance with the format defined by an intermediate representation, and LLVM bitcode obtained by disassembling binary code, are both types of intermediate code and can also be regarded as intermediate code in abbreviated form. The code composition content of the “risk code” described herein includes any binary code for loading and execution, intermediate code, and/or any computer program code written in a computer programming language. The types of the risk code include a patch code, a malicious program code, and/or any code used to correct or generate a software vulnerability or a software virus. The “metadata” described herein, also known as interpretive data, intermediary data, or metadata, refers to data describing an object or information related to the object. The “risk code login list” described herein may include any list logged, registered, and/or represented by a structure declared by a structured language and/or a structured protocol. The list may be stored and published in the form of a file, a database, and/or a web page. The content of the list includes login information of the risk code described herein, where the login information includes the content of the risk code itself, metadata of the risk code, software vulnerability information or software virus information corresponding to the risk code; where the CVE (Common Vulnerabilities and Exposures) list (CVE Repository) is an implementation form of the risk code login list described herein, but is not limited thereto. The “semantic information” described herein may include values retrieved from memory and/or intermediate variables during the simulation execution of the code, the execution history of each instruction during execution, output/input values of functions, and/or any retrievable numerical information. All or part of the semantic information of two codes can be used as component features for comparison between the two, and the similarity between the two codes is determined according to the degree of consistency in the comparison.

[0074] In addition, the “code sending module”, “update record generating module”, “scanning result notification module”, “first feature extraction unit”, “second feature extraction unit”, “feature comparison unit”, “analysis unit”, “retrieval unit”, and “detection unit” described herein are processing units that provide logical computing or services. The implementation forms of the processing units may include a class library, a software component, a software agent program, an application program, any program block that can be built-in, installed, introduced, or extended to run in the “device-code-based risk scanning system or computing device”

described herein, or any integrated circuit unit designed in a hardware description language and can be embedded in the “computing device for device-code-based risk scanning” described herein.

[0075] The “device-code-based risk scanning system” described herein may be implemented in forms including a system firmware, an application program, a software engine, a website system, an operating system, or any software platform that can be extended by programs. In a preferred embodiment, the “device-code-based risk scanning system” described herein can be deployed to run on the same machine, that is, the code sending module, update record generating module, scanning result notification module, first feature extraction unit, second feature extraction unit, feature comparison unit, analysis unit, retrieval unit, and detection unit described herein are all integrated into the same system, and the system is deployed to run on a machine. In another preferred embodiment, the “device-code-based risk scanning system” described herein can be implemented through the collaboration of multiple machines, that is, the processing units such as the code sending module, update record generating module, scanning result notification module, first feature extraction unit, second feature extraction unit, feature comparison unit, analysis unit, retrieval unit, and detection unit described herein are provided to different machines. The processing units in different machines collaborate and interact with each other through connection approaches including the Internet, WiFi, Bluetooth, etc., and are further jointly integrated into a device-code-based risk scanning system through multiple machines.

[0076] The “computing device for device-code-based risk scanning” described herein may be implemented in forms including a network communication device, a server, or any information and communication device with networking, data processing, or control capabilities. In a preferred embodiment, the “computing device for device-code-based risk scanning” described herein can be implemented by a single machine, that is, the code sending module, update record generating module, scanning result notification module, first feature extraction unit, second feature extraction unit, feature comparison unit, analysis unit, retrieval unit, and detection unit described herein are all integrated into the same machine as the computing device for device-code-based risk scanning. In another preferred embodiment, the “computing device for device-code-based risk scanning” described herein can be implemented through the collaboration of multiple machines, that is, the processing units such as the code sending module, update record generating module, scanning result notification module, first feature extraction unit, second feature extraction unit, feature comparison unit, analysis unit, retrieval unit, and detection unit described herein are provided to different machines. The processing units in

different machines can collaborate and interact with each other through connection approaches including the Internet, WiFi, Bluetooth, etc., and are further jointly integrated into a computing device for device-code-based risk scanning through multiple machines.

[0077] As mentioned above, in a preferred embodiment, the “code sending module”, “update record generating module”, “scanning result notification module”, “first feature extraction unit”, “second feature extraction unit”, “feature comparison unit”, “analysis unit”, “retrieval unit”, and “detection unit” described herein can be individually implemented in different processing units. In another preferred embodiment, at least two of the “code sending module”, “update record generating module”, “scanning result notification module”, “first feature extraction unit”, “second feature extraction unit”, “feature comparison unit”, “analysis unit”, “retrieval unit”, and “detection unit” described herein can also be implemented in the same processing unit.

[0078] The content of the present invention will be further described in detail below with reference to the accompanying drawings and specific implementations.

[0079] Referring to FIG. 1, which is a schematic flow diagram of a device-code-based risk scanning method provided by an embodiment of the present invention, specifically including the following steps:

[0080] S1. Obtaining a device-code, and sending the device-code from a first device 5 to a second device 6; where the device-code is acquired from the first device 5 and is generated by the first device 5, and includes at least one software code;

[0081] S2. Generating an update record; where the update record includes a relationship record between the device-code and the first device 5 and/or the second device 6; and

[0082] S3. Sending, when a security risk is detected in the device-code, a scanning result to the first device 5 and/or the second device 6 according to the update record.

[0083] As shown in FIG. 2, which is a structural block diagram of a device-code risk scanning system 1 of the present invention, providing an explanation of a preferred embodiment of the device-code risk scanning method of the present application. The risk scanning system 1 is a website system, and the website system can be deployed on the first device 5, the second device 6, or any computing device 11 with networking and logical computing processing capabilities; the code sending module 2, the update record generating module 3, and the scanning result notification module 4 are processing units integrated into the risk scanning system 1.

[0084] In a preferred embodiment, the risk scanning system 1 is deployed on the second device 6. The first device 5 can provide a device-code to the code sending module 2 by calling an application programming interface (API) provided by the code sending module 2 through a software system running on the first device 5, so that the code sending module 2 can obtain the device-code acquired from the first device 5 and enable the device-code to be sent from the first device 5 to the second device 6. The application programming interface may include technical implementations such as REST API and Web Service, but is not limited thereto. The implementation of the code sending module 2 itself may also include a web front-end program or a software agent program implemented in Java language. The first device 5 can dynamically load the code sending module 2 implemented as a web front-end program through a browser and execute the web front-end program on the browser; or the risk scanning system 1 provides another application programming interface for downloading the software agent program, and the first device 5 calls the another application programming interface to download and obtain the software agent program, then dynamically loads and executes the software agent program in the Java virtual environment of the first device 5. After the dynamic loading and execution of the web front-end program or the software agent program are completed, the web front-end program or the software agent program can obtain the device-code acquired from the first device 5 on the first device 5 and send the device-code from the first device 5 to the second device 6 through the application programming interface. The device-code is a device-code stored in a storage medium of the first device 5, and the web application program or the software agent program can obtain the device-code by reading the storage medium. In addition, the code sending module 2 can also directly establish a TCP/UDP Socket connection to the first device 5. For example, the code sending module 2 can access the first device 5 through a connection using the SSH (Secure Socket Shell) communication protocol as an authorized identity, traverse the storage medium and search for the device-code through system commands for accessing the storage medium, then obtain the required device-code from the storage medium of the first device 5 through a connection using the SCP (Secure Copy Protocol) communication protocol, and send the obtained device-code to the second device 6, so as to enable the device-code to be sent from the first device 5 to the second device 6.

[0085] In a preferred embodiment, the risk scanning system 1 is deployed on the first device 5. The device-code is a device-code stored in a storage medium of the first device 5. The code sending module 2 obtains the device-code by reading the storage medium, then can establish a TCP/UDP Socket connection to the second device 6, send the device-code to the

second device 6 through a mutually agreed communication protocol, and the device-code is updated as the device-code operated by the second device 6. For example: the code sending module 2 can establish a TCP/UDP Socket connection to the second device 6, access the first device 6 through a connection using the SSH (Secure Socket Shell) communication protocol as an authorized identity, obtain the required device-code from the storage medium of the first device 5 through a connection using the SCP (Secure Copy Protocol) communication protocol, send the obtained device-code to the second device 6, so as to enable the device-code to be sent from the first device 5 to the second device 6, and the device-code is updated to the second device 6, so that the device-code becomes the device-code operated by the second device 6.

[0086] In a preferred embodiment, the risk scanning system 1 is deployed on a computing device 11 different from the first device 5 and the second device 6. By adopting the same implementation manner as the above “risk scanning system 1 on the second device 6”, the risk scanning system 1 deployed on the computing device 11 can enable the code sending module 2 to obtain a device-code from the first device 5 by means of the first device 5 calling the application programming interface of the code sending module 2 and/or dynamically loading the web application program or software agent program included in the code sending module 2 into the first device 5, then establish a TCP/UDP Socket connection to the second device 6, and send the device-code to the second device 6 through a mutually agreed communication protocol. The device-code is a device-code stored in a storage medium of the first device 5.

[0087] In a preferred embodiment, an HTTP interface can be enabled on the second device 6, allowing the code sending module 2 to establish a TCP/UDP Socket connection to the HTTP interface and transmit the device-code to the second device 6 through the HTTP communication protocol.

[0088] The above manner of obtaining the device-code from the first device 5 and sending it to the second device 6 is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements. The advantage of the above technical implementation is that when the risk scanning system 1 is deployed on the second device 6, regardless of whether the first device 5 is an online server, an Internet of Things sensor, a network communication device (for example: a WiFi wireless router), or another embedded device with limited computing capabilities, the computing work of analyzing the code components of the device-code obtained from the first device 5 does not need to be executed on the first device 5; nor does it need to install additional application programs on the first device 5 to obtain the device-code. Similarly, when the risk scanning system 1 is deployed on the first

device 5, regardless of whether the second device 6 is an online server, an Internet of Things sensor, a network communication device (for example: a WiFi wireless router), or another embedded device with limited computing capabilities, the computing work of analyzing the code components of the device-code running on the second device 6 does not need to be executed on the second device 6. When the risk scanning system 1 is deployed on another computing device 11 different from the first device 5 and the second device 6, regardless of whether the first device 5 and/or the second device 6 is an online server, an Internet of Things sensor, a network communication device (for example: a WiFi wireless router), or another embedded device with limited computing capabilities, the computing work of analyzing the code components of the device-code obtained from the first device 5 and/or the computing work of analyzing the code components of the device-code running on the second device 6 do not need to be executed on the first device 5 and the second device 6.

[0089] In a preferred embodiment, the code sending module 2 in the risk scanning system 1 obtains a device-code from the first device 5 in response to an update request.

[0090] In a preferred embodiment, the risk scanning system 1 runs on the second device 6. The update request is an update request generated on the first device 5, and when the update request is generated, version information or device specification information is provided into the update request. Solution 1: The update request is generated by a software system running on the first device 5 itself. When the update request is generated, version information or device specification information is provided into the update request. The software system obtains a device-code corresponding to the version information or the device specification information from the storage medium of the first device 5 according to the version information or the device specification information in the update request, then the software system calls the application programming interface provided by the code sending module 2 to provide the device-code to the code sending module 2, or the software system calls an application programming interface provided by the risk scanning system 1 for submitting an update request to submit the update request to the code sending module 2 of the risk scanning system 1. After that, the code sending module 2 on the second device 6 accesses the first device 5 through a connection using the SSH (Secure Socket Shell) communication protocol as an authorized identity, finds the device-code by traversing the storage medium through system commands for accessing the storage medium according to the version information or the device specification information in the update request, and obtains the required device-code from the storage medium of the first device 5 through a connection using the SCP (Secure Copy Protocol) communication protocol. Solution

2: The update request is generated by a user operating on a human-machine interface of a dynamically loaded web front-end program or software agent program running on the first device 5 (for example: operating an interface for establishing and submitting an update request). When the update request is generated, the user can input version information or device specification information into the update request through the human-machine interface. After submitting the update request, the dynamically loaded web front-end program or software agent program can obtain a device-code corresponding to the version information or the device specification information from the storage medium of the first device 5 according to the version information or the device specification information in the update request, and send the device-code to the second device 6.

[0091] In a preferred embodiment, the risk scanning system 1 runs on the first device 5. The update request is sent from the second device 6 to the first device 5, and when the update request is generated, version information or device specification information is provided into the update request. Solution 1: A software system running on the second device 6 generates the update request, fills version information of the required device-code or device specification information of the second device 6 into the update request, and sends the update request to the code sending module 2 of the risk scanning system 1 running on the first device 5 by calling an application programming interface provided by the risk scanning system 1 for submitting an update request. After obtaining the update request, the code sending module 2 can obtain the device-code corresponding to the version information or the device specification information from the storage medium of the first device 5, then send the device-code to the second device 6, and update the device-code as the device-code running on the second device 6. Solution 2: The update request can also be generated by the risk scanning system 1 running on the first device 5. When the update request is generated, version information or device specification information is provided into the update request. After the code sending module 2 obtains the device-code corresponding to the version information or the device specification information from the storage medium of the first device 5 according to the version information or the device specification information in the update request, the code sending module 2 can send the device-code to the second device 6, and update the device-code as the device-code running on the second device 6.

[0092] In a preferred embodiment, the risk scanning system 1 runs on another computing device 11 different from the first device 5 and the second device 6. The update request is an update request generated on the first device 5, and when the update request is generated, version information or device specification information is provided into the update request; or the update

request is an update request generated by the second device 6 and is sent to the first device 5, and when the update request is generated, version information or device specification information is provided into the update request. Solution 1: The update request is generated by a software system running on the first device 5 itself or by a user through a dynamically loaded web front-end program or software agent program of the code sending module 2. By calling an application programming interface provided by the risk scanning system 1 for sending device-code, the device-code corresponding to the version information or the device specification information is sent to the another computing device 11, and then sent to the second device 6 by the code sending module 2, and updated to the device-code running on the second device 6. Solution 2: The update request is generated by a software system running on the first device 5 itself or by a user through a dynamically loaded web front-end program or software agent program of the code sending module 2. By calling an application programming interface provided by the risk scanning system 1 for submitting an update request, the update request is submitted to the code sending module 2 of the risk scanning system 1. The code sending module 2 accesses the first device 5 through a connection using the SSH (Secure Socket Shell) communication protocol as an authorized identity, finds the device-code by traversing the storage medium through system commands for accessing the storage medium according to the version information or the device specification information in the update request, obtains the required device-code from the storage medium of the first device 5 through a connection using the SCP (Secure Copy Protocol) communication protocol, then sends the device-code to the second device 6 by the code sending module 2, and updates the device-code as the device-code running on the second device 6. Solution 3: A software system running on the second device 6 generates the update request, fills version information of the required device-code or device specification information of the second device 6 into the update request, and sends the update request to the code sending module 2 of the risk scanning system 1 running on the first device 5 by calling an application programming interface provided by the risk scanning system 1 for submitting an update request. After obtaining the update request, the code sending module 2 obtains the device-code corresponding to the version information or the device specification information from the storage medium of the first device 5, then sends the device-code to the second device 6, and updates the device-code as the device-code running on the second device 6.

[0093] In a preferred embodiment, both the version information and the device specification information are part of the metadata of the device-code. The device specification information includes at least one of a product specification, a hardware specification, and a

firmware specification of a device on which the device-code is supported to operate.

[0094] The advantage derived from the technical means by which the above update request is generated by the first device 5 is that the second device 6 does not need to know the network location information of the first device 5 and/or the another computing device 11. The first device 5 and the another computing device 11 complete the update of the device-code for the second device 6 in a push manner. In a preferred embodiment, the network location information of the second device 6 is filled into the update request by the user through the human-machine interface of the dynamically loaded web front-end program or software agent program, or the network location information of the second device 6 is written into the update request by the risk scanning program 1 after the risk scanning program 1 receives the update request; then the code sending module 2 transmits the device-code to the second device 6 for update according to the network location information in the update request. This way of completing the update of the device-code in an asynchronous manner, the sending of the update request and the receiving of the device-code are completed via two connections with different connection objects, which is different from the traditional OTA update where the second device 6 itself establishes a connection to the server providing the update code for downloading.

[0095] In fact, the above scenario makes the update of the device-code unnecessary for the second device 6 itself to download. With permission, both the first device 5 and the another computing device 11 can send an update request to the risk scanning system 1 through the above improved approach. Subsequently, the risk scanning system 1 establishes a connection to the second device 6 indicated by the network location information according to the network location information of the second device 6, and then actively transmits the device-code to the second device 6 to complete the update.

[0096] The network location information can be simply an IP Address, and there is a default communication protocol (a Port supported by the device by default) between the risk scanning system 1 and the second device 6 for transmitting the device-code. In this case, the risk scanning system 1 only needs to know the IP Address of the device to actively connect to it and then transmit the updated device-code. In addition, the network location information can also be an IP Address plus a Port, that is, the network location information includes specified information of the network communication protocol. In this way, there is no need for any default network communication protocol between the risk scanning system 1 and the second device 6, and the risk scanning system 1 can also establish a connection between the first device 5 and the first device 5 or the another computing device 11 according to the IP Address (location) plus the

Port (communication protocol) and transmit the updated device-code. The approach of completing the update of the device-code in an asynchronous manner is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0097] There are multiple ways for the risk scanning system 1 to obtain the network location information of the second device 6. Solution 1: A user of the second device 6 registers and logs the metadata of the second device 6 into a database of the risk scanning system 1, where the metadata of the second device 6 includes the network location information of the second device 6. The risk scanning system 1 can find the network location information of the second device 6 by searching the metadata of the second device 6 in the database; the metadata of the second device 6 can also include at least one of the product specification, hardware specification, and firmware specification of the second device 6, and the risk scanning system 1 can search the database through the information included in the update request (such as the product specification, hardware specification, and firmware specification supported by the device-code). Solution 2: The risk scanning system 1 and the second device 6 are in a network environment where they can interact through Multicast packets. The risk scanning system 1 can periodically send "Multicast packets for device search" to the network. When the second device 6 is a device that supports the communication protocol of the Multicast packets, after receiving the Multicast packets, the second device 6 can write its own network location information and at least one of its product specification, hardware specification, and firmware specification into a responding Multicast packet and send it to the network environment. After receiving it, the risk scanning system 1 can grasp the network location information, product specification, hardware specification, and firmware specification of the second device 6 and log them into the database.

[0098] In a preferred embodiment, the risk scanning system 1 runs on the first device 5 or another computing device 11. When the code sending module 2 of the risk scanning system 1 obtains an update request, in addition to obtaining the required device-code according to the version information and device specification information (at least one of product specification, hardware specification, and firmware specification) included in the update request, it can also search the above-mentioned database through the device specification information to locate the network location information of the second device 6, then send the device-code to the second device 6, so as to update the device-code running on the second device 6.

[0099] In a preferred embodiment, the code sending module 2, the update record generating module 3, and the scanning result notification module 4 are regarded as processing

units integrated into the risk scanning system 1. Each processing unit can provide at least one function to realize interaction with other processing units. When one processing unit needs to send a request to another processing unit, the content of the request can be used as a parameter, and the content of the request can be input to the other processing unit in the form of a function call. After receiving the content of the request, the other processing unit can provide services for the request according to the content of the request. In another preferred embodiment, each processing unit can also realize interaction with other processing units through network connection, Inter-process communication, or other approaches. When one processing unit needs to send a request to another processing unit, the processing unit can establish a TCP/UDP Socket connection to the other processing unit, transmit the content of the request to the other processing unit through a mutually agreed communication protocol, or send the content of the request to the other processing unit through an inter-process communication mechanism implemented by partial memory sharing, so that the other processing unit can receive the content of the request and provide services for the request according to the content of the request; where the implementation approach of how to provide a request to a processing unit and how the processing unit receives the request is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0100] In another preferred embodiment, the update request is directly generated by the risk scanning system 1 and then submitted to the code sending module 2. The risk scanning system 1 can generate the update request according to the content of a configuration file and submit it to the code sending module 2. The content of the configuration file can include relevant information required for obtaining a device-code from the first device 5 and sending the device-code to the second device 6 as described above. The relevant information includes, but is not limited to, the network location information of the first device 5 and/or the second device 6, the authorized identity of the first device 5 and/or the second device 6, version information, and/or device specification information. The configuration file can be set for the risk scanning system 1 by the administrator of the risk scanning system 1 when deploying the risk scanning system 1, or provided to the risk scanning system 1 by the administrator and/or its users of the risk scanning system 1 through operational settings on the human-machine operation interface of the risk scanning system 1 when the risk scanning system 1 is already operating online.

[0101] Based on the above description, the generation and transmission of the content of the update request, and the approach for the code sending module 2 to obtain the update request are not limited thereto, and those skilled in the art can make any equivalent changes and designs

according to actual application requirements.

[0102] In a preferred embodiment, after the code sending module 2 obtains the update request, in addition to transmitting the device-code from the first device 5 to the second device 6, it can further store the device-code in a database or a file server and request the update record generating module 3 to establish an update record in the database of the risk scanning system 1. Solution 1: The risk scanning system 1 runs on the first device 5. In addition to the version information of the device-code, the device specification information supported by the device-code, and the storage location information of the device-code in the file server, the update record further includes a relationship record, where the relationship record includes the metadata of the second device 6. Solution 2: The risk scanning system 1 runs on the second device 6. In addition to the version information of the device-code, the device specification information supported by the device-code, and the storage location information of the device-code in the file server, the update record further includes a relationship record, where the relationship record includes the metadata of the first device 5. Solution 3: The risk scanning system 1 runs on another computing device 11 different from the first device 5 and the second device 6. In addition to the version information of the device-code, the device specification information supported by the device-code, and the storage location information of the device-code in the file server, the update record further includes a relationship record, where the relationship record includes the metadata of the first device 5 and the metadata of the second device 6. The metadata of the first device 5 includes the network location information of the first device 5 and the account information of the user, operator, and/or manager of the first device 5. The metadata of the first device 5 may include the network location information of the first device 5 and the account information of the user, operator, and/or manager of the first device 5.

[0103] In a preferred embodiment, the first device 5 and/or the second device 6 is an embedded system device, a server device, an Internet of Things sensing device, a network communication device, or any computer device with networking capabilities.

[0104] Further, the device-code is acquired from the first device 5, and the at least one software code included in the device-code includes at least one binary code.

[0105] The device-code risk scanning method enables the risk scanning system 1 to obtain the device-code from the first device 5 and further update the device-code of the second device 6, making the device-code the code executed by the second device 6 during online operation; when the first device 5 and the second device 6 are offline, the risk scanning system 1

implementing the device-code risk scanning method can perform risk scanning of the device-code for the first device 5 and the second device 6, that is, by analyzing the code components of the device-code stored in the file server. With the above method, when the first device 5 and the second device 6 are online, the risk scanning system 1 can also perform risk scanning of the device-code for the first device 5 and the second device 6, that is, both online and offline situations are involved in the device-code risk scanning method.

[0106] In a preferred embodiment, the device-code obtained from the first device 5 is acquired in the form of a software code package, that is, the device-code is packaged into a software code package; the form of the software code package may include a compressed file containing all software codes included in the device-code. In another preferred embodiment, the device-code is acquired in the form of a system image file, and the system image file contains all software codes included in the device-code.

[0107] In a preferred embodiment, regardless of whether in the form of a system image file or a software code package, after the code sending module 2 of the risk scanning system 1 obtains the device-code, all software codes included in the device-code can be scanned. To perform the scan, the system image file containing the device-code is mounted to a memory or a sector divided from the hard disk memory (when it is a zImage, its content can be decompressed into a folder by means of decompression), or the software code package containing the device-code is decompress into a specified folder of the hard disk memory, and then the complete content of the device-code can be stored in a file server. The files and directories of the device-code stored on the file server is traversed, which is equivalent to scanning the component composition contained in the system image file and the software code package. After grasping the file and directory structure of the device-code according to the traversal result, it can be compared with the file and directory structures of known operating system cores, driver programs, application programs, common software components, and/or class libraries to find out the operating system cores, driver programs, application programs, common software components, and/or class libraries with matching file and directory structures. According to the above procedure, the risk scanning system 1 can scan and analyze a composition list of the software codes included in the device-code after collecting the device-code. The composition list may include metadata of each software code existing in forms such as an operating system core, a driver program, an application program, a common software component, and/or a class library (the metadata is data describing information related to the software code; for example, descriptive information such as the name of each software code can be a form of metadata), and

the content of the composition list may also include metadata of the device-code itself (the metadata is data describing information related to the device-code).

[0108] In a preferred embodiment, according to the composition list of the at least one software code included in the device-code, the risk scanning system 1 can monitor the CVE list corresponding to each software code contained in the composition list (for example, for the Linux Kernel, monitor the CVE list corresponding to the Linux Kernel), and detect code analysis events by monitoring abnormal changes in the CVE list. When a vulnerability is added to the list, a patch code corresponding to the vulnerability is provided and added to the vulnerability record, or the patch code of the vulnerability is updated, leading to abnormal changes in the content of the list, a code analysis event can be triggered; where the patch code is a form included in the risk code, but is not limited thereto. For the malicious software code which is one form of the risk code, those skilled in the art can make any equivalent changes and designs according to actual application requirements. The code analysis event corresponding to the risk code is triggered when the risk code is added or updated, and the triggered event can be received by the risk scanning system 1. The CVE list is an implementation form of the risk code login list described herein, but is not limited thereto, and those skilled in the art can make any equivalent changes and designs for different forms of risk code login lists according to actual application requirements.

[0109] In a preferred embodiment, when a patch code is added or updated, thereby triggering a code analysis event corresponding to the patch code, the risk scanning system 1 can retrieve the patch code from the storage medium storing the patch code; where the storage medium may be the same as the file server storing the device-code, or may be another storage medium. In another preferred embodiment, when the risk scanning system 1 monitors that a patch code is added or updated, it can first retrieve and compile the source code of the patch code stored in another storage medium (for example: GIT Hub where the patch code is released), and store the compiled result (binary code or intermediate code) in the file server. In this case, the pre-compiled result can be regarded as part of the entire patch code, and during component comparison, the risk scanning system 1 retrieves the entire patch code (including source code, binary code, and/or intermediate code) from the file server and the another storage medium. The above approach for monitoring the risk code login list and retrieving codes is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0110] In a preferred embodiment, after the risk scanning system 1 retrieves the patch

code, it can compare the component contents of the patch code and the corresponding at least one software code in the device-code. When the comparison result indicates that the at least one software code contains at least one security risk, the scanning result notification module 4 can be requested to send a scanning result regarding the security risk.

[0111] In a preferred embodiment, the scanning result includes scanning status notification information and/or a scanning report. Sending the scanning result refers to sending the scanning status notification information and/or the scanning report to at least one of a user, an operator, or a manager of the first device 5 and the second device 6 via at least one network communication channel, such as email, mobile phone push notification, mobile phone short message notification, web push, and other network communication channels; where the scanning report is used to show a result of comparing component contents of the at least one software code with the at least one risk code. The patch code is one form included in the risk code. The existence of the patch code itself indicates that there is a corresponding known or newly discovered vulnerability in the software code included in the device-code, and the patch code is the code for correcting the vulnerability. When the comparison result of the component contents of the patch code and the software code shows that the component content of the software code does not include the patch code, it means that the vulnerability still exists in the device-code. Unscrupulous persons can use the vulnerability to attack the first device 5 and/or the second device 6 that execute the device-code and are running the device-code online. That is, the existence of the vulnerability is a risk to the first device 5 or the second device 6. Based on this, the component content comparison result showed in the content of the scanning report may include “description information of the at least one risk code”, “description information of the at least one software code”, “description information of the device-code”, “description information of the vulnerability”, “description information of the risk”, “the number of vulnerabilities contained in the device-code and/or the at least one software code”, “the security level of the device-code and/or the at least one software code”, and/or “a determination result of whether the at least one software code includes the at least one risk code”, but is not limited thereto. The scanning status notification information is a notification message sent to relevant persons regarding the completion status of the scanning result, that is, providing a notification of the scanning status to at least one of the user, operator, or manager of the first device 5 and the second device 6 via at least one network communication channel regarding the completion status of the scanning result. The content of the scanning status notification information may include “description information of the scanning report and/or its content” and/or “the complete or

partial content of the scanning report”, but is not limited thereto.

[0112] In a preferred embodiment, when the form of the risk code is a malicious software code, the malicious software code represents the code of a known vulnerability. That is, when the comparison result of the component contents of the malicious software code and the software code shows that the component content of the software code still includes the malicious software code, it means that the vulnerability still exists in the device-code, and a scanning result needs to be sent.

[0113] In a preferred embodiment, the scanning result notification module 4 can retrieve the update record of the device-code from the database of the risk scanning system 1, search for a relationship record from the content of the update record, and send the scanning result to at least one of the user, operator, or manager of the first device 5 and/or the second device 6 through at least one user account information of at least one of the user, operator, or manager of the first device 5 and/or the second device 6 recorded in the relationship record. That is, according to the at least one user account information, confirming the email address and/or mobile phone number of at least one of the user, operator, and/or manager, and sending the scanning result (scanning report and/or scanning status notification information) to the user, operator, and/or manager via network communication channels such as email, mobile phone push notification, and mobile phone short message notification.

[0114] In a preferred embodiment, when the user, operator, and/or manager uses the web front-end program of the risk scanning system 1 on the first device 5 or the second device 6, the scanning result notification module 4 can send the scanning result from the back-end to the web front-end program through the web front-end program.

[0115] The content of the above scanning result, scanning report, and scanning status notification information, as well as the sending approach of the scanning result, are not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0116] In a preferred embodiment, before the risk scanning system 1 obtains the device-code, it can read the content of at least one risk code login list (for example: CVE lists of different software projects), find the release source of the patch code (for example: GIT Hub for releasing patch codes) according to the vulnerability records contained in the content of the at least one risk code login list, retrieve and compile the source code of the patch code stored in another storage medium (for example: GIT Hub for releasing patch codes), and store the

compiled result (binary code or intermediate code) in the file server. Through this technical implementation, at least one patch code (an implementation form of the risk code) corresponding to the at least one risk code login list is pre-collected in the file server. In this way, when the composition list of the at least one software code included in the device-code is obtained by analyzing the device-code, the risk scanning system 1 can locate the risk code login list corresponding to each software code in the composition list according to each software code in the composition list, and then retrieve the pre-collected risk code according to the corresponding risk code login list from the file server to compare the component contents with each software code in the device-code, so as to obtain the scanning result for the entire device-code. As can be seen from the above, the code analysis event can be triggered in different scenarios, including at least: 1) triggered when a composition list of the at least one software code included in the device-code is obtained; 2) triggered when at least one risk code is added to the corresponding risk code login list; or 3) triggered when at least one risk code is updated in the corresponding risk code login list, but is not limited thereto. Those skilled in the art can make any equivalent changes and designs according to actual application requirements. In addition, it can be seen that the at least one risk code can be pre-collected into the file server according to at least one risk code login list, or collected when a risk code is added to or updated in the at least one risk code login list. However, the approach of collecting risk codes is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0117] Regarding the “component content of the software code” and the “component content of the risk code” described herein, the “component content” refers to including “program instructions contained in the code composition content of the code”, “storage structure of the code in the memory”, and “environment settings and configuration data required for executing the code”; where the code composition content of the code includes any binary code for loading and execution, intermediate code, and/or any computer program code written in a computer programming language. In addition, the “security risk” described herein includes the risk that the code contains vulnerabilities, viruses, and/or malicious software, and/or any risk that causes the code to perform unexpected logical computing. The above approach of scanning for a security risk by comparing the risk code with the software code is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0118] As shown in FIG. 1, the code component analysis before detecting a security risk

when the security risk is detected in the device-code in S3 specifically includes the following steps:

[0119] S31. Extracting at least one first component feature corresponding to the risk code;

[0120] S32. Extracting at least one second component feature corresponding to the at least one software code; and

[0121] S33. Comparing the at least one first component feature with the at least one second component feature.

[0122] Further, the first component feature described in S31 is obtained by analyzing code composition content and/or a code execution result corresponding to the risk code; where the code composition content includes the risk code and/or a first intermediate code obtained by compiling the risk code, and the code execution result includes semantic information revealed during simulation execution of the risk code and/or the first intermediate code when performing simulation execution on the risk code and/or the first intermediate code.

[0123] Further, the at least one second component feature described in S32 is obtained by analyzing at least one code composition content and/or at least one code execution result corresponding to the at least one software code; where the at least one code composition content includes at least one binary code and/or at least one second intermediate code obtained by disassembling the at least one binary code, and the at least one code execution result includes semantic information revealed during simulation execution of the at least one binary code and/or the at least one second intermediate code when performing simulation execution on the at least one binary code and/or the at least one second intermediate code.

[0124] Further, a programming language structure of the first intermediate code and/or the at least one second intermediate code is declared by an intermediate language, an intermediate presentation, a structured language, and/or a structured protocol.

[0125] In a preferred embodiment, types of the risk code include a patch code associated with the at least one software code, a malicious program code, and/or any code used to correct or generate a software vulnerability or a software virus.

[0126] In a preferred embodiment, the code analysis event of the risk code is triggered when the risk code is added to or updated in a risk code login list (for example: a CVE list).

[0127] In a preferred embodiment, before the code analysis event is triggered, the

method further includes: detecting at least one code analysis event by the risk scanning system 1 according to a composition list of the at least one software code. The code analysis event corresponding to the risk code is one of the at least one code analysis event.

[0128] In a preferred embodiment, the component composition of the device-code may include software codes existing in different forms such as a system core, a driver program, an application program, a component, and/or a class library. The code composition content of each software code may include some Scripts, binary codes, and/or intermediate codes. Therefore, the at least one software code can be understood as part of the device-code. The at least one code composition content can be understood as part of the at least one software code.

[0129] Generally speaking, the at least one software code contained in a device includes a list, for example: which may include components such as Linux Kernel, OpenSSL, sqlite, and the like. As these components are continuously inspected, vulnerabilities are discovered and corresponding patch codes are released, which are then added to or updated in online CVE lists (CVE repositories) and GIT Hub.

[0130] When these vulnerabilities and patch codes are added to or updated in online CVE lists and GIT Hub by the respective entities or organizations, the risk scanning system 1 described herein will not be notified. Therefore, for the code analysis event to be triggered, it means that before the trigger, the risk scanning system 1 should monitor CVE lists and GIT Hub such as Linux Kernel, OpenSSL, sqlite, etc., according to the composition list of the device-code, so as to detect these code analysis events.

[0131] The technical effect achieved thereby is that the risk scanning system 1 can immediately complete the code security detection of the device-code when a vulnerability and a patch code are added to or updated in an online risk code login list (for example: a CVE list).

[0132] Further, the composition list is obtained by the risk scanning system 1 scanning and/or analyzing the component composition of the device-code when the device-code is provided to the risk scanning system 1.

[0133] The device-code may be a single .o file (object file), .img file (system image file), .so file (program library), or a .zip or .gz compressed file (the content includes all .o and .so files required by the entire device, and the path structure of these files is also retained in the compressed file).

[0134] For the overall device-code of a device, there are many files of software codes

included therein. Each software code is combined in the device-code in the form of different components (for example, Linux Kernel, OpenSSL, sqlite, etc.). The files of each software code are located in different path positions. There are different ways to obtain the composition list of the software codes included in the device-code. One way is to scan the entire device-code for file path information. As software codes of different components, their related files are placed in different positions, so it can be basically determined which components are contained in the device-code. In addition, the content of each software code file can also be read to check whether there are certain characteristics of specific components in the file content to grasp the component list. For example, when reading the code composition content in a code file included in a software code, it can be compared with known code segments of known software such as Linux Kernel, OpenSSL, sqlite, etc. According to the comparison result, it can be determined which components are contained in the device-code. The approach of scanning and/or analyzing the component composition of the device-code to obtain the composition list of each software code in the device-code is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0135] In a preferred embodiment, the code composition content of the patch code includes a file written in Javascript and containing Javascript code. The function of the patch code is to correct a vulnerability of a software code included in the device-code. The risk scanning system 1 can first locate the position of the Javascript file corresponding to the file name in the code composition content of the software code by means of the file name of the Javascript file. After reading the contents of the Javascript file of the patch code and the corresponding Javascript file of the software code respectively, the component contents contained in the two files are compared, that is, it is compared whether the contents of the two files include the corrected code segment. The corrected code segment is used as the component feature for comparison. A consistent comparison result indicates that the software code does not contain the vulnerability, and an inconsistent comparison result indicates that the software code contains the vulnerability.

[0136] In a preferred embodiment, in addition to segments contained in the code as features, the component features for comparison can also be the storage structure of the patch code and the software code in the memory and/or the environment settings and configuration information of the patch code and the software code. For example: a new file is added to the code composition content of a patch code, and the file name of this file does not exist in the code composition content of the corresponding software code; and/or a new parameter is added to the

environment settings and configuration of the patch code, and this parameter does not exist in the environment settings and configuration of the corresponding software code. It can be said that the component content of the software code does not include the patch code, and thus the vulnerability corresponding to the patch code is still present in the software code.

[0137] As in the above example, the code segments, file name structure, and/or configuration content of the patch code used for comparison are the first component features in the component content comparison. The code segments, file name structure, and/or configuration content of the software code used for comparison are the second component features in the component content comparison. The approach of comparing component features obtained by analyzing code composition content is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0138] In a preferred embodiment, the code composition content of a software code includes a first file containing binary code, and the file name of the first file and its relative position in the storage structure can correspond to a second file contained in a patch code and written in C/C++ source code. The second file contains code for correcting a vulnerability. In this case, the first file is first disassembled into a first intermediate code (code in the form of LLVM IR as the intermediate code representation), and the second file is also compiled and then disassembled into a second intermediate code (code in the form of LLVM IR as the intermediate code representation). Furthermore, simulation execution can be performed on the first intermediate code and the second intermediate code, that is, by reading the program instructions in the first intermediate code and the second intermediate code, feeding in simulated parameters, and performing logical computing according to the read program instructions. After all program instructions are executed, the semantic information revealed during the simulation execution process can be obtained according to the execution history of each instruction during the execution of the program instructions, the output and input of function calls, and/or the execution order; where the “semantic information” described herein includes the values of memory and/or intermediate variables during program operation, and/or the instructions, functions called during the simulation execution process, and their output and input values. A similarity is obtained by comparing the semantic information of the first intermediate code with the semantic information of the second intermediate code, and it can be determined whether the code segments of the first intermediate code and the second intermediate code are consistent according to the similarity, thereby identifying whether the first file contains the code for correcting the vulnerability in the second file. When the component features formed by all the contents of the semantic information

of the first intermediate code and the semantic information of the second intermediate code are completely included in semantic information of each other, the similarity can be said to be 100%. When the component features formed by only a part of the contents are included in the semantic information of each other, the proportion of the part to the total content is the similarity. Based on this, a similarity result ranging from 0% to 100% can be obtained. When the similarity exceeds a set threshold, it indicates that the first file contains the code of the second file. The threshold is a reference value for determining whether the code of the second file is included in the first file. For example, when the threshold is set to 90%, it means that when the similarity is higher than 90%, the first file contains the code of the second file; otherwise, the code of the second file is not contained in the first file.

[0139] In a preferred embodiment, the code composition content of a software code includes a first file containing intermediate code (Java bytecode), and the file name of the first file and its relative position in the storage structure can correspond to a second file contained in a patch code and written in Java source code. The second file contains code for correcting a vulnerability. In this case, after compiling the second file into intermediate code (Java bytecode), its semantic information is obtained through simulation execution, and then it can be compared whether the first file contains the code for correcting the vulnerability in the second file through the component features such as values retrieved from memory and/or intermediate variables contained in the semantic information, and/or instructions, functions called during the simulation execution process, and their output and input values.

[0140] The above approach of comparing component features of semantic information obtained through simulation execution is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0141] As shown in FIG. 5, the present embodiment further provides a device-code risk scanning computing device 11, including a code sending module 21, an update record generating module 31, and a scanning result notification module 41; where

[0142] the code sending module 21 is configured to obtain a device-code, where the device-code is acquired by a first device 5, and is sent to a second device 6 from the first device 5; where the device-code includes at least one software code;

[0143] the update record generating module 31 is configured to generate an update record; where the update record includes a relationship record between the device-code and the first device 5 and/or the second device 6; and

[0144] the scanning result notification module 41 is configured to send a scanning result to the first device 5 and/or the second device 6 according to the update record when a security risk is detected in the device-code.

[0145] When the scanning result indicates that the at least one software code contains at least one security risk, the scanning result notification module 41 is further configured to send the scanning result. The scanning result notification module 41 sends the scanning result to at least one of a user, an operator, or a manager of the first device 5 and the second device 6 via at least one network communication channel.

[0146] In a preferred embodiment, the computing device 11 is an embedded device. The code sending module 21, the update record generating module 31, and/or the scanning result notification module 41 are designed in a hardware description language and can be integrated into an integrated circuit unit of the computing device 11 or into an embedded system of the computing device 11 in the form of software code. The code sending module 21, the update record generating module 31, and the scanning result notification module 41 are driven by the embedded system running in the computing device 11.

[0147] In another preferred embodiment, the computing device 11 is a server. The code sending module 21, the update record generating module 31, and/or the scanning result notification module 41 are integrated into a software system in the form of software code. The software system is provided to the computing device 11 for operation through a built-in or installation manner, and the code sending module 21, the update record generating module 31, and the scanning result notification module 41 are driven by the software system running in the computing device 11.

[0148] In another preferred embodiment, the computing device 11 includes distributed servers. The code sending module 21, the update record generating module 31, and/or the scanning result notification module 41 are integrated into a distributed software system in the form of software code. The distributed software system is deployed to the distributed servers for operation to form the computing device 11. The code sending module 21, the update record generating module 31, and the scanning result notification module 41 are driven by the distributed software system running in the computing device 11.

[0149] The above manner of composing the computing device 11 through combinations of different forms of hardware and software is not limited thereto, and those skilled in the art can make any equivalent changes and designs according to actual application requirements.

[0150] Further, with reference to FIG. 2, the risk scanning system 1 includes a first feature extraction unit, a second feature extraction unit, and a feature comparison unit; where

[0151] the first feature extraction unit is configured to extract at least one first component feature corresponding to the at least one risk code;

[0152] the second feature extraction unit is configured to extract at least one second component feature corresponding to the at least one software code; and

[0153] the feature comparison unit is configured to compare the at least one first component feature with the at least one second component feature.

[0154] Still further, the first feature extraction unit or the second feature extraction unit includes at least one analysis unit and at least one retrieval unit; where

[0155] the at least one analysis unit is configured to obtain the at least one first component feature by analyzing code composition content and/or a code execution result corresponding to the at least one risk code, and/or obtain the at least one second component feature by analyzing code composition content and/or a code execution result corresponding to the at least one software code; and

[0156] the at least one retrieval unit is configured to extract the at least one first component feature or the at least one second component feature obtained after analysis.

[0157] With reference to FIG. 3, the device-code risk scanning computing device 11 further includes at least one detection unit; where

[0158] the at least one detection unit is configured to monitor a risk code login list (for example: a CVE list) corresponding to each software code contained in the composition list according to the composition list of the at least one software code included in the device-code, and detect a code analysis event by monitoring abnormal changes in the risk code login list.

[0159] In summary, the present invention discloses a device-code risk scanning method, system, and computing device 11. By acquiring the device-code of an online running device, updating the device-code of an online running device, comparing the component contents of the risk code and the device-code, and sending scanning results for the devices, risk scanning can be performed on the devices regardless of whether the devices are offline, the network status is normal, or the computing capability is insufficient, thereby realizing risk prevention and control for the device-code.

[0160] The above detailed descriptions are specific explanations of feasible embodiments of the present invention. The embodiments are not intended to limit the scope of protection of the present invention. All equivalent implementations or changes made without departing from the present invention shall be included in the scope of protection of the present invention.

WHAT IS CLAIMED IS:

1. A device-code-based risk scanning method, characterized by comprising:

obtaining a device-code, and sending the device-code from a first device to a second device; wherein the device-code is acquired from the first device and comprises at least one software code;

generating an update record; wherein the update record comprises a relationship record between the device-code and the first device and/or the second device; and

sending, when a security risk is detected in the device-code, a scanning result to the first device and/or the second device according to the update record.

2. The device-code-based risk scanning method according to claim 1, characterized in that sending the scanning result refers to sending a notification of the scanning result to at least one of a user, an operator, or a manager of the first device and the second device via at least one network communication channel.

3. The device-code-based risk scanning method according to claim 1, characterized in that the at least one software code comprises at least one binary code.

4. The device-code-based risk scanning method according to claim 1, characterized in that obtaining the device-code specifically comprises:

acquiring the device-code from a storage medium of the first device according to an update request.

5. The device-code-based risk scanning method according to claim 4, characterized in that an approach for obtaining the device-code further comprises:

obtaining, from the storage medium and when the update request is acquired, a device-code corresponding to version information or device specification information according to the version information or the device specification information in the update request as the device-code; wherein the device specification information comprises at least one of a product specification, a hardware specification, and a firmware specification of a device.

6. The device-code-based risk scanning method according to claim 4, characterized in that sending the device-code to the second device specifically comprises:

sending the device-code to the second device according to network location information of the second device contained in the update request.

7. The device-code-based risk scanning method according to claim 1, characterized in that the security risk is detected in the device-code specifically comprises:

comparing, when at least one code analysis event is triggered, component contents of at least one risk code with the at least one software code; wherein the at least one risk code comprises at least one of a patch code associated with the at least one software code, a malicious program code, and any computer program code written in a computer programming language.

8. The device-code-based risk scanning method according to claim 7, characterized in that the at least one risk code is pre-collected according to at least one risk code login list, or collected when a risk code is added to or updated in the at least one risk code login list.

9. The device-code-based risk scanning method according to claim 7, characterized in that the at least one code analysis event is triggered when a composition list comprising the at least one software code is obtained, or triggered when the at least one risk code is added or updated.

10. The device-code-based risk scanning method according to claim 7, characterized in that comparing the at least one risk code with the at least one software code specifically comprises:

extracting at least one first component feature corresponding to the at least one risk code;

extracting at least one second component feature corresponding to the at least one software code; and

comparing the at least one first component feature with the at least one second component feature.

11. The device-code-based risk scanning method according to claim 10, characterized in that the at least one first component feature is obtained by analyzing code composition content and/or a code execution result corresponding to the at least one risk code.

12. The device-code-based risk scanning method according to claim 11, characterized in that the code composition content comprises the at least one risk code and/or at least one first intermediate code obtained by compiling the at least one risk code, wherein the code execution result comprises semantic information revealed during simulation execution of the at least one risk code and/or the at least one first intermediate code when performing simulation execution on the at least one risk code and/or the at least one first intermediate code.

13. The device-code-based risk scanning method according to claim 10, characterized in that the at least one second component feature is obtained by analyzing at least one code composition content and/or at least one code execution result corresponding to the at least one software code.

14. The device-code-based risk scanning method according to claim 13, characterized in that the at least one code composition content comprises at least one binary code and/or at least one second intermediate code obtained by disassembling the at least one binary code, wherein the at least one code execution result comprises semantic information revealed during simulation execution of the at least one binary code and/or the at least one second intermediate code when performing simulation execution on the at least one binary code and/or the at least one second intermediate code.

15. The device-code-based risk scanning method according to claim 12 or claim 14, characterized in that a programming language structure of the intermediate code is declared by an intermediate language, an intermediate representation, a structured language, and/or a structured protocol.

16. The device-code-based risk scanning method according to claim 7, characterized by further comprising, before the code analysis event is triggered:

detecting at least one code analysis event according to a composition list of the at least one software code, wherein the code analysis event corresponding to the at least one risk code is one of the at least one code analysis event.

17. The device-code-based risk scanning method according to claim 9, characterized in that the composition list is obtained by scanning and/or analyzing a component composition of the device-code when the device-code is acquired.

18. The device-code-based risk scanning method according to claim 17, characterized in that the composition list comprises metadata of the device-code itself and/or metadata of the at least one software code contained in the component composition of the device-code.

19. The device-code-based risk scanning method according to claim 18, characterized in that the at least one software code is present in the component composition of the device-code in forms comprising a system core, a driver program, an application program, a component, and/or a class library.

20. The device-code-based risk scanning method according to claim 17, characterized by comprising:

monitoring a risk code login list corresponding to each software code contained in the composition list according to the composition list, so as to detect a code analysis event by monitoring the abnormal changes in the risk code login list.

21. The device-code-based risk scanning method according to claim 17, characterized in that the approach for obtaining the composition list specifically comprises:

scanning and/or analyzing, when the device-code is a system image file, a component composition contained in the system image file to identify the at least one software code present in the system image file, so as to obtain the composition list of the at least one software code; or

scanning and/or analyzing, when the device-code is a software code package, a component composition contained in the software code package to identify the at least one software code present in the software code package, so as to obtain the composition list of the at least one software code.

22. The device-code-based risk scanning method according to claim 1, characterized in that sending the scanning result to the first device and/or the second device according to the update record specifically comprises:

searching the update record to obtain the relationship record in the update record; and

sending the scanning result to at least one of a user, an operator, or a manager of the first device and the second device according to the relationship record.

23. The device-code-based risk scanning method according to claim 22, characterized in that the scanning result comprises at least one of a scanning report and scanning status notification information; wherein the scanning report is used to show a result of comparing component contents of the at least one software code with the at least one risk code; the scanning status notification information is used to provide a notification of a scanning status to at least one of a user, an operator, or a manager of the first device and the second device.

24. A device-code-based risk scanning system, characterized by comprising: a code sending module, an update record generating module, and a scanning result notification module; wherein

the code sending module is configured to obtain a device-code, and send the device-code from a first device to a second device; wherein the device-code is acquired from the first device and comprises at least one software code;

the update record generating module is configured to generate an update record; wherein the update record comprises a relationship record between the device-code and the first device and/or the second device; and

the scanning result notification module is configured to send a scanning result to the first device and/or the second device according to the update record when a security risk is detected in the device-code.

25. The device-code-based risk scanning system according to claim 24, characterized by further comprising: a first feature extraction unit, a second feature extraction unit, and a feature comparison unit; wherein

the first feature extraction unit is configured to extract at least one first component feature corresponding to at least one risk code;

the second feature extraction unit is configured to extract at least one second component feature corresponding to the at least one software code; and

the feature comparison unit is configured to compare the at least one first component feature with the at least one second component feature.

26. The device-code-based risk scanning system according to claim 25, characterized in that the first feature extraction unit or the second feature extraction unit comprises at least one analysis unit and at least one retrieval unit; wherein

the at least one analysis unit is configured to obtain the at least one first component feature by analyzing code composition content and/or a code execution result corresponding to the at least one risk code, and/or obtain the at least one second component feature by analyzing code composition content and/or a code execution result corresponding to the at least one software code; and

the at least one retrieval unit is configured to extract the at least one first component feature and/or the at least one second component feature obtained after analysis.

27. The device-code-based risk scanning system according to claim 24, characterized by further comprising: at least one detection unit; wherein

the at least one detection unit is configured to monitor a risk code login list corresponding to each software code contained in a composition list according to the composition list of the at least one software code, so as to detect a code analysis event by monitoring abnormal changes in the risk code login list.

28. A computing device for device-code-based risk scanning, characterized by comprising: a code sending module, an update record generating module, and a scanning result notification module; wherein

the code sending module is configured to obtain a device-code, and send the device-code from a first device to a second device; wherein the device-code is acquired from the first device and comprises at least one software code;

the update record generating module is configured to generate an update record; wherein the update record comprises a relationship record between the device-code and the first device and/or the second device; and

the scanning result notification module is configured to send a scanning result to the first device and/or the second device according to the update record when a security risk is detected in the device-code.