

METHOD AND SYSTEM OF DECENTRALIZED MALWARE IDENTIFICATION

The technical solution relates to a system and a method for ensuring cybersecurity, namely, to a method for decentralized malware check and a system implementing this method.

Background

The term "blockchain" is widely used today to describe the technology of decentralized transaction confirmation that uses formation of a sequential chain of blocks. In this case, the transaction is a minimal logically meaningful data processing operation that makes sense and may only be performed as one whole. A transaction block is a special structure to record a group of transactions in the Bitcoin system and in similar ones. The transaction is considered as completed and confirmed one when its format and encrypted digital signatures have been verified, and when the transaction itself has been combined into a group with several other transactions and written into a special structure – a block. The blocks may be created by members ("miners") of a peer-to-peer network in the scope of which the blockchain technology is implemented. The blockchain network nodes use a consensus protocol that ensures consistency of the network register contents and ensures that the distributed registers of the peer-to-peer network nodes are the exact copies.

The block contents may be checked since each block contains information about the previous block. All blocks are lined up in one chain that contains information about all operations ever performed in the database [<https://en.wikipedia.org/wiki/Blockchain>]. Modifying a block causes all subsequent blocks to become invalid. The blocks with unconfirmed or incomplete transactions may also be recognized as invalid ones. A chain containing a certain number of invalid blocks may also be recognized as an invalid chain. Each block contains a header and the transactions themselves. The header contains hash values: the header hash, the previous block hash, the transaction hashes and other service information. Hash algorithms (for example, SHA-256) ensure that any small change in transaction input data will result in a different hash value as a result. The blockchain network decentralization deprives individual members and member groups of the opportunity to control the system.

Usage of the blockchain is known to store and validate domain names (see, for example, US2016191243). Distributed responsibility and decentralization that are the blockchain technology base provide protection against errors in the domain name database. In particular,

the described technical solution does not allow to "contaminate" the DNS cache that stores potentially false information about unauthorized resolvers. This eliminates the danger of redirecting requests of legitimate applications to network hosts exhibiting malicious activity. DNS responses usually do not have a cryptographic signature. In the described technical solution, a DNS extension is proposed that provides support of cryptographic signatures (keys) for authorization in a trusted peer-to-peer network. The signatures of each organization in the DNS are validated through a "chain of trust", including verification of parent and child keys.

In patent application US2017295157, the blockchain technology is used to identify user devices. When connecting to the network, information about the user device is requested that may be encapsulated in a block of transactions for subsequent identification of the user. The device information may contain the device settings, MAC address, installed applications and services, connected devices, and other characterizing information.

The blockchain may be used to secure Internet resources (see US2017324738). In this case, the blockchain may include domain registrars, domain name services, autonomous systems, etc. as peer-to-peer network members. The set Internet security rules may be transmitted through the peer-to-peer network that cannot be individually controlled by a single network member.

From the patent application US2017279818, a technical solution is known, including a method to use a distributed register (in particular, the blockchain system) implemented on a computer that implements verification and distribution of virus signatures. These signatures may be used by antivirus engines in computing devices to detect and remove malware. Antivirus specialists (miners) who have access to the distributed register can test new signatures. The distributed register and antivirus applications may be configured to block new signatures until they are checked by some specified minimum number of the miners.

The method described in the application US2017279818 includes access through a computer system to data related to virus signatures. Each signature corresponds to a signature rating based on the number of the miners who had added and verified the signature. The computer system can determine authenticity of the virus signature based on the signature rating. If the signature is viral indeed, it may be used by a computer system, for example, for virus scanning.

The specified distributed register, according to the patent application US2017279818 description, may be a distributed database configured to implement the blockchain technology. An electronic register copy may be stored and maintained by the computer system. The other copies may be stored and maintained by other computer systems that are nodes of the computer network. A computer system may receive via a network the

transactions that are sent by other computer systems and show addition of new virus signatures to the computer network or notify about the results of signature verification by other computer systems in the network.

The ways to check the malware signatures are known that use centralized confirmation of the signature authenticity. For example, in the US7730040 patent, malware detection is based on user feedback. Since such a system uses a centralized malware database, the possibility of its changes remains.

The present technical solution is aimed to improve the accuracy and reliability of identifying malware.

Disclosure

The first object is a decentralized malware check method, the method executed in a computer device having access to a distributed register and a transaction pool that are implemented in a peer-to-peer network, the method comprising:

- receiving input data related to a potentially malware;
- checking the potentially malware based on the received data related to the potentially malware;
- adding check parameters and at least one result of the potentially malware check into the transaction pool;
- receiving results of the distributed check of the potentially malware from a set of peer-to-peer network computer devices;
- determining the harmfulness parameter based on the results of a distributed check of the potentially malware received from the set of peer-to-peer network computer devices;
- identifying the malware in response to the harmfulness parameter of the potentially malware exceeds the predetermined threshold value;
- saving the identified malware and associated data related to the malware and received during execution of the method in the distributed malware register.

In one embodiment of the method, the input data is received from at least one of the following sources: the computer devices in this peer-to-peer network, at least one client device, the pre-formed database, a remote server, computer-readable media.

In another embodiment the input data contains at least one pointing to the potentially malware file.

In yet another embodiment, the input data additionally comprises at least one of the following: the malware signature, the data on the malware attribution.

In yet another method embodiment, the input data is received at least partially in a hashed form.

In yet another method embodiment, at least a portion of the input data is hashed in response to the fact that at least the specified portion of the input data has been received in a non-hashed form.

In yet another method embodiment, after receiving the input data, at least a portion of the additional input data related to this malware is extracted from at least one additional source accessible to the computer device.

In yet another method embodiment, the potentially malware is checked taking into account the additional input data related to this potentially malware.

In yet another method embodiment, the potentially malware is checked using the machine-learning algorithm.

In yet another method embodiment, the training sample of the machine-learning algorithm is updated additionally in accordance with the confirmed results.

In yet another method embodiment, the potentially malware is checked in an automated way.

In yet another method embodiment, the potentially malware is checked manually.

In yet another method embodiment, the potentially malware check based on the data related to this potentially malware comprises at least one of the following:

- determining harmfulness of the potentially malware;
- validating the signature of the potentially malware;
- determining data on attribution of the potentially malware.

In yet another method embodiment, the data on attribution of a potentially malware is determined on the basis of the data that is related to the potentially malware and received at least from one of the following sources:

- received input data;
- distributed malware register;
- malicious resource database.

In yet another method embodiment, the data on attribution of a potentially malware is determined using the machine-learning algorithm.

In yet another method embodiment, at the stage of inputting at least one result of the potentially malware check into the pool of transaction parameters and results, the additional data, which is related to the potentially malware and received at least from one additional source available to a computer device, are input into the transaction pool too.

In yet another method embodiment, the harmfulness parameter is determined based on the results of the distributed potentially malware check that are received from the set of peer-to-peer network computer devices, and on at least one of the following parameters:

number of the peer-to-peer network computer devices that downloaded data about the indicated potentially malware;

reputation of the peer-to-peer network computer devices that downloaded data about the indicated potentially malware;

number of the peer-to-peer network computer devices that confirmed the result of the check of the indicated potentially malware;

reputation of the peer-to-peer network computer devices that confirmed the result of the check of the indicated potentially malware.

In yet another method embodiment, after inputting the potential malware check results into the distributed register, the tokens are charged to the computer devices of the distributed peer-to-peer network that were used to get the check results that correspond to the harmfulness parameter determined based on the results of the distributed check of the potentially malware received from the peer-to-peer network computer devices.

In yet another method embodiment, after saving the identified malware and associated data that is related to the malware and received during the method execution, into the distributed malware register, the malware and its associated data, which is related to the malware and received during the method execution, are additionally saved into at least one malware database accessible to a computer device.

In yet another method embodiment, after saving the identified malware and associated data associated data that is related to the malware and received during the method execution, into the distributed malware register, the malware and its associated data, which is related to the malware and received during the method execution, are additionally saved into the learning set of the machine-learning algorithm.

Another object of this technique is a **computer device** for decentralized malware check that has access to the distributed register and the transaction pool implemented on the basis of the peer-to-peer network, the computer device comprises a communication interface and at least one computer processor functionally connected with the communication interface, the computer processor configured to:

receive input data related to a potentially malware;

check the potentially malware based on the received data related to the potentially malware;

add check parameters and at least one result of the potentially malware check into the transaction pool;

receive results of the distributed check of the potentially malware from a set of peer-to-peer network computer devices;

determine the harmfulness parameter based on the results of a distributed check of the potentially malware received from the set of peer-to-peer network computer devices;

identify the malware in response to the harmfulness parameter of the potentially malware exceeds the predetermined threshold value;

save the identified malware and associated data related to the malware and received during execution of the method in the distributed malware register.

In one possible embodiment of the device, the processor is additionally configured to receive input data from at least one of the following sources: computer devices in this peer-to-peer network, at least one client device, a pre-formed database, a remote server, a computer readable media.

In another embodiment, the processor is additionally configured to receive input data including at least one pointing to the potentially malware file.

In yet another device embodiment, the processor is additionally configured to receive input data additionally comprising at least one from the following:

- malware signature;

- malware attribution data.

In yet another device embodiment, the processor is additionally configured to receive input data at least partially in a hashed form.

In yet another device embodiment, the processor is additionally configured to hash at least a portion of the input data in response to the fact that at least the specified portion of the input data has been received in a non-hashed form.

In yet another device embodiment, at least one additional source is available, and the processor is additionally configured to extract at least a portion of the additional input data related to the malware from at least one additional source.

In yet another device embodiment, the processor is additionally configured to check the potentially malware taking into account the additional input data related to this potentially malware.

In yet another device embodiment, the processor is additionally configured to check the potentially malware using the machine-learning algorithm.

In yet another device embodiment, the processor is additionally configured to update the training sample of the machine-learning algorithm in accordance with the confirmed results.

In yet another device embodiment, the processor is additionally configured to check the potentially malware in an automated way.

In yet another device embodiment, the processor is additionally configured to check the potentially malware manually.

In yet another device embodiment, the processor is additionally configured to check the potentially malware based on the received input data related to this potentially malware and to execute one of the following functions:

- determining harmfulness of the potentially malware;

- validating the signature of the potentially malware;
- determining data on attribution of the potentially malware.

In yet another device embodiment, the processor is additionally configured to determine the data on attribution of a potentially malware based on the data that is related to the potentially malware and received at least from one of the following sources:

- received input data;
- distributed malware register;
- malicious resource database.

In yet another device embodiment, the processor is additionally configured to determine the data on attribution of a potentially malware using a model based on neural networks.

In yet another device embodiment, the processor is additionally configured to add into the transaction pool the additional data that is related to the potentially malware and received at least from one additional source available to the computer device.

In yet another device embodiment, the processor is additionally configured to determine the harmfulness parameter, based on the results of the distributed potentially malware check that are received from the set of the peer-to-peer network computer devices, and on at least one of the following parameters:

- number of the peer-to-peer network computer devices that downloaded data about the indicated potentially malware;

- reputation of the peer-to-peer network computer devices that downloaded data about the indicated potentially malware;

- number of the peer-to-peer network computer devices that confirmed the result of the check of the indicated potentially malware;

- reputation of the peer-to-peer network computer devices that confirmed the result of the check of the indicated potentially malware.

In yet another device embodiment, the processor is additionally configured to store the malware and the associated data related to the malware into at least one malware database accessible to the computer device.

In yet another device embodiment, the processor is additionally configured to store the malware and the associated data related to the malware into the learning set of the machine-learning algorithm.

In the context of this description, unless otherwise is specifically stated, the words "**first**", "**second**", "**third**", etc. are used as adjectives solely to distinguish nouns to which they refer from each other, but not for the purpose to describe any particular relationship between these nouns.

In the context of this description, unless otherwise is specifically stated, a "**malware**" is any software designed to gain unauthorized access to computing resources or information for the purpose of illegally using resources or causing harm (damage) to the information owner or network by copying, distorting, deleting or substituting information.

In the context of this description, unless otherwise is specifically stated, a "**potentially malware**" is any software that is suspicious for one reason or another and has been selected to be checked in order to confirm its harmfulness. In the scope of this technology, any software that has not been checked may be considered as a potentially malware.

In the context of this description, unless otherwise is specifically stated, a "**computer device**" is any hardware and/or software suitable to solve the corresponding task. Thus, some non-limiting examples of computer devices may include computer processors, computer systems (one or more servers, desktops, notebooks, netbooks, etc.), smart phones, tablets, etc.

In the context of this description, unless otherwise is specifically stated, a "**peer-to-peer network node**" is a software that is implemented on a computer device, is connected to other peer-to-peer network nodes and is capable to perform the operations described below.

In the context of this description, unless otherwise is specifically stated, a "**transaction pool**" is a distributed database containing transactions formed in one of peer-to-peer network nodes that is used to implement the distributed register as well as the specified transactions not confirmed by other peer-to-peer network nodes.

In the context of this description, unless otherwise is specifically stated, a "**pointing**" to a file may be an information element itself or a pointer, reference, link or other indirect way that allows the recipient of the pointing to find a network, memory, database or other computer-readable medium from which the information element can be extracted. For example, a pointing to a malware file may contain a universal resource locator (for example, the URL www.xyz.com/worm.exe) identifying the malicious file with respect to a particular network (in particular, the Internet), or may by some other means transfer to the recipient an pointing to a network folder, memory address, database table or another place where you can access the malicious file, its individual fragments or signature. As will be understood by those skilled in the art, the accuracy degree required for such pointing depends on the degree of primary understanding the way to interpret the information exchanged by the receiver and sender of the pointer. For example, if, before data transfer between the sender and recipient, it is clear that the information element pointer takes the form of a universal resource pointer URL, all that is necessary to transfer video content effectively to the recipient is transfer of the pointer or link to this video content, despite the information element (for example, a malicious file, its

fragment or signature) itself was not transferred between the sender and recipient of the pointing.

In the context of this description, unless otherwise is specifically stated, an "**attribution**" of a malware is definition of information (attributes) relating to persons, systems and/or organizations that have developed or use the specified malware or its part. In the context of this description, unless otherwise is specifically stated, a "**signature**" is a set of rules to identify malicious activity. For example, a malware signature is referred to any signs that characterize uniquely the specified malware. Signatures also include, for example, rules to detect malicious traffic, behavioral rules for "sandboxes", YARA rules.

In the context of this description, unless otherwise is specifically stated, a "**authentic signature**" of a malware is a signature for which the peer-to-peer network members determined and confirmed that the specified signature characterizes the malware uniquely.

In the context of this description, unless otherwise is specifically stated, a "**transaction**" is a minimal logically meaningful data processing operation that makes sense and may only be performed as one whole.

In the context of this description, unless otherwise is specifically stated, the expression "**deep neural network**" implies a system of programs and data structures created for the approximate modeling of processes in a human brain. The deep neural networks generally include a series of algorithms that can identify the underlying relationships and connections in a data set using a process that mimics the human brain function. Locations and weights of the data set links generally determine the output. Thus, the deep neural network is generally open for all input data or parameters simultaneously, in its entirety, and, therefore, is able to model their interdependence. Unlike machine-learning algorithms that use decision trees and, therefore, have their limitations, the deep neural networks are not limited and, therefore, are suitable to model interdependencies.

In the context of this description, unless otherwise is specifically stated, a "**manual check**" is a manual or partially automated check, requiring the operator participation.

In the context of this description, a "**computer device**" is a computer program operating on relevant equipment that is able to receive requests (for example, from client devices) over the network and execute these requests or initiate execution of these requests. The equipment may be one physical computer or one physical computer system, but neither the one nor the other is necessary for this technical solution. In the context of this technical solution, use of the expression "computer device" does not mean that each task (for example, received commands or requests) or any particular task will be received, executed or initiated to be executed by the same "computer device" (i.e. the same software and/or hardware), but

it means that any number of software elements or hardware devices may be involved in receiving/transmitting, executing or initiating the execution of any request or consequence of any request related to the client device, and all this software and hardware may be a single computer device or several computer devices, both options are comprised in the expression "at least one computer device".

In the context of this technical description, a “**server**” is a computer program that is operating on the appropriate hardware and configured to receive requests (for example, from computer devices) over the network and to execute these requests, or to cause these requests to be sent. This hardware may be one physical computer or one physical computer system, but none of them is required as an option, in accordance with this technology. In this context, use of the expression “server” does not mean that each task (for example, instructions received) or any particular task will be received, executed, or will cause execution by the same server (i.e. by the same program and/or hardware), but it means that any number of software elements or hardware devices may be involved in receiving/sending, executing, or may cause execution of any task or request, or of consequences of any task or request; and all this software and hardware may be implemented as one or more servers.

In the context of this description, unless otherwise is specifically stated, a “**token**” is an object of the blockchain technology, performing cryptographically protected confirmation of its owner rights to receive valuables promised to this owner or of ability to execute predetermined functions using this object.

In the context of this description, a “**machine-readable medium**” is a medium of absolutely any type and nature, including RAM, ROM, disks (CDs, DVDs, floppy disks, hard disks, etc.), USB flash drives, solid-state drives , tape drives, etc.

In the context of this description, a “**database**” is any structured data set that does not depend on a particular structure, database management software, computer hardware where data is stored, used or otherwise available to be used. The database may be located in the same hardware that executes the process to store or use the information stored in the database, or it may be located in a separate hardware, for example, a dedicated server or several servers.

In the context of this description, unless otherwise is specifically stated, by “**external database**” is a database related with a computer device in the peer-to-peer network, through computer devices that are not in the specified peer-to-peer network.

In the context of this description, unless otherwise is specifically stated, a “**result weight**” is the numerically expressed degree of authenticity of the result obtained due to at

least one action with a malware or with data related to the malware. For instance, in a non-limiting example, the result weight of determining the software harmfulness may express the probability that the software is malicious that is determined by the subject of the check or is calculated on the basis of the results of several subjects of the check. The result weight of the malware signature generation may imply an indicator characterizing increase in the efficiency of malware detection when using the specified signature. Additional and/or alternative characteristics, aspects and advantages of the embodiments of this technical solution will become apparent from the following description, the accompanying drawings, and the appended claims.

Brief Description of the Drawings

To understand this technical solution better, as well as its other aspects and characteristics, a link is provided to the following description that should be used in conjunction with the accompanying drawings, where:

Fig. 1 shows a simplified schematic representation of one of the non-limiting implementations of the system embodying this technical solution.

Fig. 2 shows a flowchart of a sequence of operations according to the claimed method of decentralized malware identification.

Detailed description

The following description is presented only to describe an exemplification of the present technical solution. This description is not intended to delimit the present technology or define its scope.

Some useful examples to modify the described method and the system that determines the related network resources may also be covered by the following description. Its purpose is also solely to help in understanding, but not to determine the scope and boundaries of this technology. These modifications are not an exhaustive list, and those skilled in the art will understand that other modifications are possible. In addition, it should not be interpreted so that where it has not yet been done, i.e. where no modification examples have been provided, no modifications are possible and/or something described here is the only embodiment of this element of the present technology. As it will be clear to a person skilled in the art, this is most likely not the case. In addition, it should be kept in mind that, in some specific cases, the method and system to determine the related network resources are fairly simple embodiments of the present technology, and in such cases they are presented here in order to facilitate

understanding. As it will be clear to a person skilled in the art, many embodiments of the present technology will be much more complex.

Fig. 1 shows a schematic representation of one of the non-limiting implementations of system 100 to implement the method described in this technical solution. In one of the non-limiting implementations of this technical solution, the system 100 contains a set of computer devices 1021, 1022, ..., 102N, that are associated with the user devices, the first 1041, the second 1042 and the Nth 104N, where the first user device 1041 is associated with the computer device 1021, the second user device 1042 is associated with the computer device 1022, and the Nth user device 104N is associated with the Nth computer device 102N. Additionally, the system 100 comprises the database 106 associated with the computer device 1021. Obviously, the system 100 may contain a different number of user devices, databases, and other external sources associated with one or more of the set of computing devices 1021, 1022, ..., 102N. In one embodiment of the present technical solution, no user device may be associated with one or more computer devices from the set of computer devices.

Each of the set of the computer devices 1021, 1022, ..., 102N is connected to each other (each with each) via a communication channel to form the peer-to-peer network 150. Each of the set of the computer devices 1021, 1022, ..., 102N is a node of the peer-to-peer network 150 and is configured to receive and transmit messages represented as blocks of transactions in accordance with the blockchain technology. It should be understood that the number of nodes of the peer-to-peer network 150, namely the number of the computer devices in the set 1021, 1022, ..., 102N is not specifically limited. As an example, but not a limitation, the peer-to-peer network 150 may contain 48, 100, 3000, or any other number of nodes.

For the purposes of this technology, each node of the peer-to-peer network 150 may be implemented as a computer device 1021, 1022, ..., 102N, and each node contains the distributed malware register 108, the transaction pool 109, the machine-learning module 110, and at least one virtual machine 1121, 1122, ... 112N to execute files of potentially malware in a virtual environment.

The computer devices 1021, 1022, ..., 102N may be implemented, for example, as a Dell™ PowerEdge™ computer server with an Ubuntu Server operating system. As a non-limiting example of implementation, the computer device 1021 may have the following characteristics: the disk memory of 2TB, 4 or more CPUs for parallel operation of one or several virtual machines, RAM of 256GB or more, and at least one graphics processor to implement the machine-learning module 110. The computer devices 1022, ..., 102N may be implemented in a similar way. It is obvious that the computer devices 1021, 1022, ..., 102N

may have the same or different configuration. These characteristics are not limiting ones and are only illustrative in order to clarify this technical solution.

Given the blockchain technology that is used to implement the peer-to-peer network 150, each node comprises the same distributed malware register 108, the same transaction pool 109, and the same machine-learning module 110. The virtual machines 1121, 1122, ... 112N to execute files of potentially malware in a virtual environment may be the same or different in different nodes of the peer-to-peer network 150. Moreover, all or some of these nodes may contain several identical and/or different virtual machine images. As an example, but not a limitation, one of the virtual machines may be implemented with the following specified set of configuration attributes: Windows XP operating system, x86 (32-bit) architecture, Russian language (rus) to run potentially 32-bit malware in it. To run 64-bit potentially malware, for example, a virtual machine may be used with the following set of attributes: Ubuntu operating system, x64 (64-bit) architecture, English language (eng).

The distributed register 108 may be a database hosted on relevant physical equipment that may be one or more computer-readable media. The distributed malware register 108 of one or several nodes of the peer-to-peer network 150 may additionally be accessible (read-only) to one or several external devices. For example, according to a non-limiting embodiment of system 100, the distributed malware register 108 hosted on the computer device 1021 may be accessible by the first user device 1041 and by the database 106 of computer malware, the distributed malware register 108 hosted on the computer device 1022 may be accessible to the second user device 1042, the distributed malware register 108 hosted on the computer device 102N may be accessible to the user device 104N.

The transaction pool 109 may be a database hosted on relevant physical equipment that may be one or more computer-readable media. In one embodiment, the transaction pool 109 and the distributed malware register 108 of some nodes may be implemented using the same physical hardware. It should be noted that, in general, only the nodes of the peer-to-peer network 150 have access to the transaction pool 109. Information input by the node of the peer-to-peer network 150 to the transaction pool 109 becomes available to all nodes of the peer-to-peer network 150.

In one of the non-limiting implementations of the present technical solution, the peer-to-peer network 150 is configured to transfer transaction blocks in accordance with the blockchain technology, specified transactions reflecting information about at least one of the following: signatures of malware/potentially malware, hash sums of malware/potentially malware files, pointing to malware/potentially malware files, results of automated check of malware/potentially malware and parameters of the algorithms used in this method, results of manual check of malware/potentially malware for each node of the peer-to-peer network, the

total check result determined on the basis of results of check of malware/potentially malware by computer devices, reputation of the nodes that execute the automated check, reputation of the nodes that execute the manual check, generation parameters.

In one of the non-limiting implementations of this technical solution, in addition to the peer-to-peer network 150, each of the set of the computer devices 1021, 1022, ..., 102N may additionally be configured to transmit data in other data networks (not shown), in particular, in the Internet, a local or other data network. The first user device 1041 may be associated with the computer device 1021, for example via the data communication network 160, which may be the Internet, the data base 106 may be also accessible to the computer device 1021 via the data communication network 160 or another network (not shown). The first user device 1041 in some non-limiting embodiments of the present technical solution may be a personal computer (desktop, notebook, etc.) where the software may be installed, for example, to identify suspicious potentially malware and malicious files and to send them via the data communication network 160 as input data to the computer device 1021 that is the node of the peer-to-peer network 150.

In addition, the first user device 1041 may have access to the distributed malware register 108. If check of potentially malware by the set of the nodes of the peer-to-peer network 150 confirms the software harmfulness, then the software, its hash sum, and at least a portion of the associated malware-related data will be input into the distributed malware register 108. Thus, the first user device 1041 may receive data about malware confirmed by the set of the nodes of the peer-to-peer network 150. The potentially malware or pointing to them may be pre-stored in the database 106. The first user device 1041 may be configured to receive from the database 106 the input data related to the potentially malware.

The second user device 1042 may be connected to the computer device 1022, for example, via the local network 170 based on a wired Ethernet network or a wireless network (Wi-Fi, Bluetooth, 3g/4g/LTE, etc.).

The second user device 1042 in some non-limiting embodiments of the present technical solution may be a server or active network equipment configured to collect data about potentially malware from the set of the devices associated with the server (not numbered). At the same time, software may be installed in the second user device 1042 to check traffic from the corporate network devices and to search malicious activity or to collect information from software installed in devices included in, for example, the corporate network.

In one embodiment, the second user device 1042 may contain a device to capture network traffic and get data about potentially malware based on network traffic analysis or may have access to such device. For example, without limitations, the network traffic capture

device may be one of the following devices: L2 network switches using network traffic mirroring technology for the required network segments, such as, for example, the network traffic mirroring SPAN technology in Cisco hardware, network transparency tools also referred to as Security Delivery Platforms or Network Packet Brokers (NPB), and Test Access Points of various types, as well as proxy servers with ICAP support working within the established TCP connection, mail servers with SMTP support, etc.

In some non-limiting embodiments of the present technique, the second user device 1042 may send potentially malware via the data communication network 170 as input data to the computer device 1022, which is a node of the peer-to-peer network 150.

In addition, the second user device 1042 may have access to the distributed malware register 108. If check of potentially malware by the set of the nodes of the peer-to-peer network 150 confirms the software harmfulness, then the software will be input into the distributed malware register 108. Thus, the second user device 1042 may receive data about malware confirmed by the set of the nodes of the peer-to-peer network 150.

The N^{th} user device 104N may be associated with the computer device 102N, for example, via the local network 180 based on a wired Ethernet network or a wireless network (Wi-Fi, Bluetooth, 3g/4g/LTE, etc.)

The N^{th} user device 104N in some non-limiting embodiments may be a server for searching and analyzing cyber threats, collecting data, in particular, about potentially malware and attribution (information (attributes) related to persons, systems and/or organizations that developed or used the specified malware or its part) associated with this potentially malware. In some non-limiting embodiments, the N^{th} user device 104N may send potentially malware via the data communication network 180 as input data to the computer device 102 N, which is a node of the peer-to-peer network 150.

The N^{th} user device 104N may have access to the distributed malware register 108. If check of potentially malware by the set of the nodes of the peer-to-peer network 150 confirms the software harmfulness, then the software will be input into the distributed malware register 108. Thus, the N^{th} user device 104N may receive data about malware confirmed by the set of the nodes of the peer-to-peer network 150.

The N^{th} user device 104N may be a usual computer server. In the exemplary embodiment of this technical solution, the N^{th} user device 104N may be a Dell™ PowerEdge™ server that uses the Ubuntu Server operating system. The N^{th} user device 104N may be any other suitable hardware and/or application software, and/or system software, or their combination. In the present embodiment of the present technical solution, which does not limit its scope, the N^{th} user device 104N is a single server. In other embodiments of the present

technical solution that do not limit its scope, the functionality of the Nth user device 104N may be divided among several user devices and, for example, may be executed by several servers.

It should be note that each user device 1041, 1042, ... 104N that has access to the distributed malware register 108 may receive data about any confirmed malware that has been checked by many nodes of the peer-to-peer network 150, regardless of the source of this malware. For example, malware received by the computer device 1021 from the first user device 1041, after confirmation, will be available to the second user device 1042 and other user devices including the Nth user device 104N.

The machine-learning module 110 is configured to use a machine learning algorithm capable to check and confirm harmfulness of a potentially malware file. In some embodiments of the present technical solution, one or more machine learning algorithms may be any suitable machine-learning algorithm with a trainer or of semi-supervised learning, such as:

- Artificial neural network
- Gaussian regression process
- Decision trees
- and so on

In general, the machine-learning module 110 executes one or more machine learning algorithms to analyze potentially malware based on results of the launch of potentially malware in at least one virtual machine 1121, 1122, ... 112N.

As will be appreciated by those skilled in the art, other similar embodiments of the system 100 are possible in the scope of the above description of the system 100.

The claimed method 200 for decentralized malware check will be discussed in more detail, referring to Fig. 2. The method 200 may be executed in any of the set of the computer devices 1021 ... 102N and, more specifically, by a processor (not shown) of any said computer device in accordance with a non-limiting embodiment of system 100 shown in Fig. 1. For purposes of the present description, we consider an illustrative example where the method 200 is executed in the computer device 1021.

Step 202: Receiving input data related to a potentially malware

The method 200 begins at step 202. The computer device 1021 receives input data related to a potentially malware. In the context of this application, the potentially malware is any software that, during its execution, is executing or may execute malicious activity, carry out unauthorized access to information, illegally use, copy, distort, delete or substitute information.

The input data may be received from at least one of the following sources: computer devices in the peer-to-peer network 150, at least one client device, a pre-formed database, a remote server, computer-readable media. It should be noted that other information sources associated with at least one computing device 1021 might also be obvious for those skilled in the art.

The input data may contain at least one pointing to a potentially malware file. In one of the non-limiting embodiments of the present technique, a pointing to a malware file may contain a universal resource locator (for example, the URL www.xyz.com) identifying the malicious file with respect to a particular network (in particular, the Internet), or may by some other means transfer to the recipient an pointing to a network folder, memory address, database table or another place where you can access the malicious file, its individual fragments or signature.

The input data may additionally comprise at least one of the following: the malware signature, the malware attribution data, the metadata of the potentially malicious file, information about the files associated with the potentially malicious file. The malware signature is referred to any signs that characterize uniquely the specified malware. The files associated with the potentially malware may be, for example, the files received with the potentially malicious files from the same source and at the same time. The related files may have common features, for example, a part of the file name, a digital signature, etc. The metadata of the potentially malicious file may comprise at least the source from which the file was received, the file size, the date of the last file modification, and other similar information.

The input data may be received at least partially in a hashed form. In response to the fact that at least a portion of the input data has been received in a non-hashed form, at least a portion of the input data may be hashed.

In one of the non-limiting embodiments of the present technique, after receiving the input data, at least a portion of the additional input data related to this malware is extracted from at least one additional source accessible to the computer device 1021 of the peer-to-peer network. In this embodiment, the check of the potentially malware is carried out taking into account the additional input data related to this potentially malware.

For example, information about the file type (a trojan program) and the file source (the antivirus installed in the user end device) may be obtained from additional sources. In this case, the potentially malware will checked, for example, by an algorithm optimized to detect trojan programs. Additionally, for example, the user device hardware and software parameters may be taken into account.

The method 200 then transits to the step 204.

Step 204: Checking the potentially malware based on the received data related to the potentially malware

At the step 204, the computer device processor checks the potentially malware based on the received data related to the potentially malware.

The potentially malware check based on the data related to this potentially malware might comprise at least one of the following:

- determining harmfulness of the potentially malware;
- determining data on attribution of the potentially malware.

The harmfulness may be determined with any of the methods known from the prior art. In one of the non-limiting embodiments, the malware is started in the virtual machine 1121. The virtual machine 1121 may be specially adapted to detect malware, in particular, in this virtual machine, user applications may be installed and services are running. The parameters of the virtual machine 1121 used to check the malware might be determined based on at least one of the following: input data related to the potentially malware, settings of the malware analysis model. The specified parameters of the virtual machine 1121 may be stored in the transaction pool 109. Thus, the parameters of the virtual machine 1121 used to check the malware by the computer device 1021 might be used to check the malicious file by the other computer devices of the peer-to-peer network 150.

The data on attribution of a potentially malware may be determined based on the data related to the potentially malware and received at least from one of the following sources:

- received input data;
- distributed malware register;
- malicious resource database.

For example, the malware may be a modification of a previously known program, and the specified known program had been previously identified as malicious and had been stored in the database 106.

The data on attribution of the potentially malware may be determined using the machine-learning module 110 that uses the malware model. For example, the model may describe the characteristic features of malware developed by a known hacker group. Building the model, for example, cyber intelligence data may be used.

In one of the non-limiting embodiments of the present technique, the potentially malware is checked by the processor of the computer device 1021 using a machine-learning algorithm to analyze malware. In this case, the machine learning module 110 is used. In particular, in

some non-limiting embodiments, the machine-learning module can determine the potentially malware harmfulness and evaluate results of checks executed by the other peer-to-peer network nodes that have access to the distributed register. The malware analysis model is pre-trained on a sample of the malware-related data.

Additionally, the malware analysis model may be re-trained with an updated sample taking into account the results confirmed by the computer devices 1021 ... 102N.

The model input data is at least the input data related to the potentially malware received by the computer device 1021 of the peer-to-peer network. In one method 200 embodiment, the potentially malware is checked in an automated way. For example, the check may be performed using the machine-learning module 110. At least a part of the automatic algorithm parameters may be set manually. For example, a weight threshold for the check result of the potential malware that is sufficient to write the specified result into the distributed register 108 may be set manually.

In another method 200 embodiment, the potentially malware is additionally checked manually. The conditions to execute a manual check may be different or absent. For example, hit of the value of the weight of the total check result to a given range of weights may be used as a condition for a manual check. This total check result is a result determined on the basis of the check results received from at least a part of the computer devices 1021 ... 102N that executed the check. The check results at least of a part of the computer devices 1021 ... 102N of the peer-to-peer network may be stored, for example, in the transaction pool 109. In another non-limiting example of the method 200 implementation, a manual check may be executed regardless of the automatic check results.

The method 200 then transits to the step 206.

Step 206: Adding check parameters and at least one result of the potentially malware check into the transaction pool

At the step 206, the results and parameters of the malware check executed by the processor of the computer device 1021 are recorded in the transaction pool 109 accessible to each of the computer devices 1021 ... 102N. An example of the check parameters may be, for example, the parameters of the virtual machine 1121. Thus, each computer device from the set of the devices 1022 ... 102N may check harmfulness of the potentially malicious file using the same check parameters that are the parameters used by the computer device 1021.

In one of the non-limiting embodiments of the method 200, at the step 206, additional data related to the potentially malware received from at least one additional source available to the computer device 1021 is also entered into the transaction pool 109.

In some non-limiting examples of this embodiment, additional sources of data related to the potentially malware may comprise the database 106, an external malware database.

The method 200 then transits to the step 208.

Step 208: Receiving results of distributed check of potentially malware from at least a part of the computer devices in the peer-to-peer network

At step 208, the computer device 1021 receives results of distributed check of potentially malware from at least a portion of computer devices 1022 ... 102N. These results may be transmitted via data communication channels available to each of the computer devices 1021 ... 102N that may transmit these results in a hashed form.

The results of distributed check of potentially malware may be stored by each of the computer devices in the peer-to-peer network 150 as blocks in the transaction pool 109 in accordance with the blockchain technology.

The method 200 then transits to the step 210.

Step 210: Determining the harmfulness parameter based on the results of a distributed check of the potentially malware received from the set of peer-to-peer network computer devices

At step 212, the processor of the computer device 1021 determines the harmfulness parameter based on the results of the distributed check of potentially malware received from at least a portion of the computer devices 1022 ... 102N.

The harmfulness parameter is a function that depends on the number N of the computer devices 1021 ... 102N in the peer-to-peer network 150, of reputation of each of these devices and of the results of determination of the harmfulness of the potentially malware received by each of the computer devices [1021 ... 102N]. For example, the harmfulness parameter may be found as follows:

$$\frac{\sum_{i=0}^N ((\text{device } 102i \text{ reputation}) * (\text{device } 102i \text{ result}))}{\text{number of devices received result}}.$$

In one of the not-limiting embodiments of the method 200, the harmfulness parameter is determined based on results of distributed check of the potentially malware, which are received from a portion of the peer-to-peer network computer devices, and based on at least one of the following parameters:

number of the computer devices from the set of the computer devices 1021 ... 102N that downloaded data about the indicated potentially malware;

reputation of the computer devices from the set of the computer devices 1021 ... 102N that downloaded the data about the indicated potentially malware;

number of the computer devices from the set of the computer devices 1021 ... 102N that confirmed the result of the check of the indicated potentially malware;
reputation of the computer devices from the set of the computer devices 1021 ... 102N that confirmed the result of the check of the indicated potentially malware.

The method 200 then transits to the step 212.

Step 212: Identifying the malware in response to the harmfulness parameter of the potentially malware exceeds the predetermined threshold value

At the step 212, the computer device 1021 processor identifies the malware in response to the harmfulness parameter of the potentially malware exceeds the predetermined threshold value.

For example, the harmfulness parameter threshold value may be set to 7. In response to the calculated harmfulness parameter is 8, the corresponding potentially malware will be marked as malicious.

The method 200 then transits to the step 214.

Step 214: Saving the identified malware and associated data related to the malware and received during execution of the method in the distributed malware register

At the step 214, the computer device 1021 processor saves the malware in the distributed malware register 108. The method to save this information and the structure of the stored information related to the malware may be configured, for example, in accordance with the blockchain technology. The saved malware information of the computer devices 1021 ... 102N may be accessed through the peer-to-peer network 150. Each of the computer devices 1021 ... 102N stores an up-to-date copy of the distributed register 108.

In this case, at least one of the following may be considered the identified malware: the malware file, the hash sum of the malware file, the malware signature. Any information related to the malware and confirmed by computer devices of the peer-to-peer network 150, for example, an pointing to the malware file, the malware signature, the files related with the malware, similar malware, the malicious file metadata, the results of check by each of the computer devices of the peer-to-peer network 150, the reputation of the specified computer devices, etc. may be considered as the accompanying data related to the malware obtained during the execution of method 200.

After the step 214, the method may finish.

In one of the non-limiting embodiments of the method 200, after saving the identified malware in the distributed malware register 108, the processor of the computer device 1021

charges tokens to the computer devices of the distributed peer-to-peer network 150 where the check results were obtained that correspond to the harmfulness parameter determined on the basis of the results of the distributed check of the potentially malware that were received from the set of the computer devices in the peer-to-peer network 150.

In one of the non-limiting embodiments of the method 200, after saving the identified malware and associated data related to the malware obtained during execution of the method in the distributed malware register 108, the computer device 1021 processor additionally saves the malware and associated data related to the malware obtained during the method execution in at least one malware database accessible to the computer device, for example, in the database 106.

In one of the non-limiting embodiments of the method 200, after saving the identified malware and associated data related to the malware obtained during execution of the method in the distributed malware register 108, the computer device 1021 processor additionally saves the malware and associated data related to the malware obtained during the method execution in the learning set of the machine-learning algorithm used by the machine-learning module 110.

EXAMPLE

A non-limiting example of one of the implementations of this technical solution is provided below.

In this non-limiting example, a decentralized malware check is executed in the node 1021 of the peer-to-peer network.

For example, the following input data may be received: a malicious file with metadata (name, size, extension, creation date, change date, etc.), the web resource URL where the file was found, the data (hardware and software, logs, etc.) of the user device where the file was run, the malware YARA signature. For example, the following YARA signature [<https://xakep.ru/2011/10/31/57409/#toc03>.] was obtained:

```
rule silent_banker : banker
{
  meta:
    description = "This is just an example"
    thread_level = 3
    in_the_wild = true
    strings:
      $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
      $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
```

```

$c = "UVODFRYSIHLNWPEJXQZAKCBGMT"
condition:
$a or $b or $c
}

```

The check by an automated way involves running a malicious file in a virtual environment and applying to it a previously trained machine-learning algorithm.

After checking, the following data is written to the transaction pool of the distributed register:

- 1) File hash sum
- 2) Result of checking the potentially malware for harmfulness by the node 1021 of the peer-to-peer network
- 3) Malicious indicators including the settings of the virtual machine 1121

After receiving the results of the malware check by other nodes of the peer-to-peer network, the malware parameter is determined. The harmfulness parameter is a function that depends on the number N of the nodes [1021 ... 102N] of the peer-to-peer network, on the rating of each of the nodes, and on the results of determining harmfulness of a potentially malware received by each of the nodes.

Based on the harmfulness parameter value, harmfulness of the potentially malware is determined, and the malware is identified.

After identifying the malware, the following information is input into the distributed register:

- 1) Pointing to the malicious file
- 2) Hash sum of the malicious file
- 3) Results of predictions of each node of the peer-to-peer network with parameters (for example, weights and architecture when training a neural network)
- 4) Manual check results
- 5) Overall result obtained using the results of the distributed check as a whole based on scoring
- 6) Reputation, the number of the peer-to-peer network nodes that performed the check and the check type (automated or manual)
- 7) Unique malware signs: description of the malware behavior in an infected system; changes in the system registry made by the malware; malware-generated files.

Further, these data are entered into the external database used to detect malware, for example, the anti-virus company database.

Modifications and improvements to the above described embodiments of this technical solution will be clear to those skilled in the art. The foregoing description is provided as an example only and is not limited in any way. Thus, the scope of this technical solution is limited only by the scope of the attached claims.

Claims

1. The decentralized malware check method, the method executed in a computer device having access to a distributed register and a transaction pool that are implemented in a peer-to-peer network, the method comprising:
 - receiving input data related to a potentially malware;
 - checking the potentially malware based on the received data related to the potentially malware;
 - adding check parameters and at least one result of the potentially malware check into the transaction pool;
 - receiving results of the distributed check of the potentially malware from a set of peer-to-peer network computer devices;
 - determining the harmfulness parameter based on the results of a distributed check of the potentially malware received from the set of peer-to-peer network computer devices;
 - identifying the malware in response to the harmfulness parameter of the potentially malware exceeds the predetermined threshold value;
 - storing the identified malware and associated data related to the malware and received during execution of the method in the distributed malware register.
2. The method of claim 1, wherein the input data is received from at least one of the following sources: computer devices in the peer-to-peer network, at least one client device, a pre-formed database, a remote server, computer-readable media.
3. The method of claim 1, wherein the input data contains at least one pointing to the potentially malware file.
4. The method of claim 1, wherein the input data additionally contains at least one of the following:
 - malware signature;
 - malware attribution data.
5. The method of claim 1, wherein the input data is received at least partially in a hashed form.

6. The method of claim 1, wherein at least a portion of the input data is hashed in response to the fact that at least the specified portion of the input data is received in a non-hashed form.
7. The method of claim 1, wherein, after receiving the input data, at least a portion of the additional input data relating to this malware is extracted from at least one additional source accessible to the computer device.
8. The method of claim 7, wherein the check of the potentially malware is executed taking into account the additional input data related to this potentially malware.
9. The method of claim 1, wherein the potentially malware check is executed using the machine-learning algorithm.
10. The method of claim 9, wherein the training sample of the machine-learning algorithm is additionally updated in accordance with the confirmed results.
11. The method of claim 1, wherein the potentially malware is checked in an automated way.
12. The method of claim 11, wherein the potentially malware is checked manually.
13. The method of claim 1, wherein the potentially malware check based on the data related to this potentially malware comprises at least one of the following:
 - determining harmfulness of the potentially malware;
 - validating the signature of the potentially malware;
 - identification of data on attribution of the potentially malware.
14. The method of claim 13, wherein the data on attribution of a potentially malware is determined on the basis of the data that is related to the potentially malware and received at least from one of the following sources:
 - the received input data;
 - the distributed malware register;
 - the malicious resource database.
15. The method of claim 13, wherein the data on attribution of a potentially malware is determined using the machine-learning algorithm.

16. The method of claim 1, wherein, at the step of entering the parameters and results of at least one result of check of potentially malware into the transaction pool, additional data that is related to the potentially malware and received from at least one additional source, available to a computer device is also entered into the transaction pool.
17. The method of claim 1, wherein the harmfulness parameter is determined based on the results of the distributed potentially malware check, which are received from the set of peer-to-peer network computer devices, and on at least one of the following parameters:
- the number of the peer-to-peer network computer devices that downloaded data about the indicated potentially malware;
 - the reputation of the peer-to-peer network computer devices that downloaded data about the indicated potentially malware;
 - the number of the peer-to-peer network computer devices that confirmed the result of the check of the indicated potentially malware;
 - the reputation of the peer-to-peer network computer devices that confirmed the result of the check of the indicated potentially malware.
18. The method of claim 1, wherein, after entering the results of the potentially malicious program into the distributed register, tokens are charged to the computer devices of the distributed peer-to-peer network where the check results were obtained, corresponding to the harmfulness parameter determined based on the results of the distributed check of the potentially malware received from the set of the peer-to-peer network computer devices.
19. The method of claim 1, wherein, after saving the identified malware and associated data related to the malware obtained during execution of the method in the distributed malware register, the malware and associated data related to the malware obtained during the method execution are additionally saved in at least one malware database accessible to the computer device.
20. The method of claim 1, wherein, after saving the identified malware and associated data related to the malware obtained during execution of the method in the distributed malware register, the malware and associated data related to the malware obtained during the method execution are additionally saved into the learning set of the machine-learning algorithm.

21. A computer device for decentralized malware check, having access to a distributed register and a transaction pool that are implemented on the basis of a peer-to-peer network, the computer device contains a communication interface and at least one computer processor functionally connected to the communication interface, the computer processor configured to:

- receiving input data related to a potentially malware;
- check of the potentially malware based on the received data related to the potentially malware;
- adding check parameters and at least one result of the potentially malware check into the transaction pool;
- receiving results of the distributed check of the potentially malware from a set of peer-to-peer network computer devices;
- determining the harmfulness parameter based on the results of a distributed check of the potentially malware received from the set of peer-to-peer network computer devices;
- identification of the malware in response to the harmfulness parameter of the potentially malware exceeds the predetermined threshold value;
- storing the identified malware and associated data related to the malware and received during execution of the method in the distributed malware register.

22. The device of claim 21, wherein the processor is additionally configured to receive input data from at least one of the following sources: computer devices in the peer-to-peer network, at least one client device, a pre-formed database, a remote server, computer-readable media.

23. The device of claim 21, wherein the processor is additionally configured to receive input data comprising at least one pointing to the potentially malware file.

24. The device of claim 21, wherein the processor is additionally configured to receive input data that additionally comprise at least one of the following:

- the malware signature;
- the malware attribution data.

25. The device of claim 21, wherein the processor is additionally configured to receive input data at least partially in a hashed form.

26. The device of claim 21, wherein the processor is additionally configured to hash at least a portion of the input data in response to the fact that at least the specified portion of the input data is received in a non-hashed form.
27. The device of claim 21, wherein at least one additional source is accessible and the processor is additionally configured to retrieve at least a part of additional input data related to this malware from at least one additional source.
28. The device of claim 27, wherein the processor is additionally configured to check the potentially malware taking into account the additional input data related to this potentially malware.
29. The device of claim 21, wherein the processor is additionally configured to check the potentially malware using the machine-learning algorithm.
30. The device of claim 29, wherein the processor is additionally configured to update the training sample of the machine-learning algorithm in accordance with the confirmed results.
31. The device of claim 21, wherein the processor is additionally configured to check the potentially malware in an automated way.
32. The device of claim 31, wherein the processor is additionally configured to get results of the check of the potentially malware in a manual way.
33. The device of claim 21, wherein the processor is additionally configured to check the potentially malware based on the received data related to this potentially malware and to execute one of the following:
determining harmfulness of the potentially malware;
validating the signature of the potentially malware;
identification of data on attribution of the potentially malware.
34. The device of claim 33, wherein the processor is additionally configured to determine the data on attribution of a potentially malware based on the data that are related to the potentially malware and received at least from one of the following sources:
received input data;
distributed malware register;

malicious resource database.

35. The device of claim 33, wherein the processor is additionally configured to determine the data on attribution of a potentially malware using a model based on neural networks.
36. The device of claim 21, wherein the processor is additionally configured to add into the transaction pool the additional data that is related to the potentially malware and received at least from one additional source available to the computer device.
37. The device of claim 21, wherein the processor is additionally configured to determine the harmfulness parameter based on the results of the distributed potentially malware check, the results are received from the set of peer-to-peer network computer devices, and on at least one of the following parameters:
 - number of the peer-to-peer network computer devices that downloaded data about the indicated potentially malware;
 - reputation of the peer-to-peer network computer devices that downloaded data about the indicated potentially malware;
 - number of the peer-to-peer network computer devices that confirmed the result of the check of the indicated potentially malware;
 - reputation of the peer-to-peer network computer devices that confirmed the result of the check of the indicated potentially malware.
38. The device of claim 21, wherein the processor is additionally configured to store the malware and the associated data related to the malware into at least one malware database accessible to the computer device.
39. The device of claim 21, wherein the processor is additionally configured to store the malware and the associated data related to the malware into the learning set of the machine-learning algorithm.