

TECHNIQUES FOR DECENTRALIZED APPLICATION DISCOVERY AND SCANNING

CROSS-REFERENCE TO RELATED APPLICATIONS

[001] This application claims the benefit of U.S. Non-Provisional Application No. 18/463,716 filed on September 8, 2023, now pending. The 18/463,716 application claims the benefit of U.S. Provisional Application No. 63/487,304 filed on February 28, 2023.

The contents of the above-referenced applications are hereby incorporated by reference.

TECHNICAL FIELD

[002] The present disclosure relates generally to decentralized applications (dApps), and more specifically to cybersecurity solutions involving scanning decentralized applications.

BACKGROUND

[003] Uses of blockchain technology have been on the rise in recent years. One particular development that came about due to the advent of blockchains is the use of blockchain as one of the building blocks for the next generation of the Internet, commonly referred to as web 3.0. More specifically, blockchains are decentralized electronic ledgers built upon a network of peer-to-peer nodes. Each peer node in a blockchain network stores a copy of the blockchain such that the blockchain is not stored in only a single centralized storage. The decentralized nature of the blockchain, in combination with procedures used for validating and recording transactions, ensures that the ledger cannot be unilaterally edited by a single node. This provides reliability of the data stored on the blockchain.

[004] The blueprint for blockchain design has been adopted as part of the plan for web 3.0. In particular, blockchain has provided a new design for a decentralized data storage which can be adapted for decentralized web design. As part of this decentralized web design, the concept of a decentralized application (dApp) has been conceived. These dApps are computer software programs which are executed over blockchain networks rather than on conventional centralized servers. Unlike previously developed computer programs, dApps are not necessarily managed or controlled by a single central authority. Instead, they are decentralized and can be collectively managed by many nodes which are owned

or operated by different entities. The blockchain protocols provide an agreed upon set of rules for these entities with respect to the blockchain that does not require any single trusted entity in order to be implemented.

[005] Existing dApp plans generally include a frontend and a backend, where the frontend is a portion of code executed on the user side (e.g., on a user device interacting with the blockchain network) and the backend is a portion of code executed on the nodes of the blockchain network. The backend may be realized as one or more smart contracts running on the nodes. The use of smart contracts distributed among multiple decentralized nodes makes the application decentralized as contrasted with running the backend code on a single central server or a set of centralized servers.

[006] New potential advantages over centralized applications are enabled through the use of dApps. For example, dApps may provide reduced or eliminated downtime due to the lack of reliance on a single server or relatively small group of servers, prevention of censorship or other central control over free exchange of data, trustless computation that does not require the owners of the nodes or the entities interacting with the dApp to trust the entities running the dApp, privacy due in part to the ability to anonymize users, data integrity due to the properties of the blockchain (e.g., that the blockchain cannot be unilaterally modified by a malicious entity), and more.

[007] Alongside these advantages, however, come certain challenges. One such challenge will be in securing this new technology against potential malicious attacks. Accordingly, cybersecurity solutions for protecting dApps and systems relying on such dApps from malicious cyber behavior are desirable.

SUMMARY

[008] A summary of several example embodiments of the disclosure follows. This summary is provided for the convenience of the reader to provide a basic understanding of such embodiments and does not wholly define the breadth of the disclosure. This summary is not an extensive overview of all contemplated embodiments, and is intended to neither identify key or critical elements of all embodiments nor to delineate the scope of any or all aspects. Its sole purpose is to present some concepts of one or more embodiments in a simplified form as a prelude to the more detailed description that is presented later. For

convenience, the term “some embodiments” or “certain embodiments” may be used herein to refer to a single embodiment or multiple embodiments of the disclosure.

[0009] Certain embodiments disclosed herein include a method for decentralized application scanning. The method comprises: generating a plurality of first transactions based on a plurality of second transactions, wherein the plurality of second transactions is from a digital wallet; broadcasting the generated plurality of transactions to a website, wherein the website returns a plurality of responses to the plurality of first transactions; extracting a plurality of calls from the plurality of responses returned by the website; and determining whether each extracted call is an anomalous call by analyzing the extracted plurality of calls.

[0010] Certain embodiments disclosed herein also include a non-transitory computer readable medium having stored thereon causing a processing circuitry to execute a process, the process comprising: generating a plurality of first transactions based on a plurality of second transactions, wherein the plurality of second transactions is from a digital wallet; broadcasting the generated plurality of transactions to a website, wherein the website returns a plurality of responses to the plurality of first transactions; extracting a plurality of calls from the plurality of responses returned by the website; and determining whether each extracted call is an anomalous call by analyzing the extracted plurality of calls.

[0011] Certain embodiments disclosed herein also include a system for decentralized application scanning. The system comprises: a processing circuitry; and a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to: generate a plurality of first transactions based on a plurality of second transactions, wherein the plurality of second transactions is from a digital wallet; broadcast the generated plurality of transactions to a website, wherein the website returns a plurality of responses to the plurality of first transactions; extract a plurality of calls from the plurality of responses returned by the website; and determine whether each extracted call is an anomalous call by analyzing the extracted plurality of calls.

[0012] Certain embodiments disclosed herein include the method, non-transitory computer readable medium, or system noted above or below, wherein determining whether each extracted call is an anomalous call results in determining at least one anomalous call

among the plurality of extracted calls, further including or being configured to perform the following steps: performing at least one mitigation action with respect to the at least one anomalous call.

[0013] Certain embodiments disclosed herein include the method, non-transitory computer readable medium, or system noted above or below, further including or being configured to perform the following steps: identifying the website as a malicious website based on the determined at least one anomalous call, wherein the at least one mitigation action is performed when the website is identified as a malicious website.

[0014] Certain embodiments disclosed herein include the method, non-transitory computer readable medium, or system noted above or below, wherein the digital wallet is a first digital wallet, further including or being configured to perform the following steps: deploying a module to a second digital wallet, wherein the module is configured to intercept requests to the second digital wallet, wherein the plurality of first transactions is broadcast via the second digital wallet, wherein the plurality of responses from which the plurality of calls is extracted is received by the second digital wallet.

[0015] Certain embodiments disclosed herein include the method, non-transitory computer readable medium, or system noted above or below, wherein the module is configured to transmit only a subset of types of transactions and simulated responses from among a set of potential types of transactions and simulated responses.

[0016] Certain embodiments disclosed herein include the method, non-transitory computer readable medium, or system noted above or below, wherein the subset of types of transactions and simulated responses includes remote procedure calls as the only input/output operation which the module is configured to send to websites.

[0017] Certain embodiments disclosed herein include the method, non-transitory computer readable medium, or system noted above or below, wherein the module is a binary large object, further including or being configured to perform the following steps: pushing an updated version of the binary large object to the second wallet, wherein the updated version of the binary large object includes differential data indicating differences from a prior version of the binary large object, wherein the plurality of transactions is broadcast via the second digital wallet using the updated version of the binary large object.

[0018] Certain embodiments disclosed herein include the method, non-transitory computer readable medium, or system noted above or below, further including or being configured to perform the following steps: determining whether each extracted call requests an anomalous transaction, wherein each extracted call which requests an anomalous transaction is determined to be an anomalous call.

[0019] Certain embodiments disclosed herein include the method, non-transitory computer readable medium, or system noted above or below, further including or being configured to perform the following steps: identifying the website based on data indicating locations of newly created websites, wherein the plurality of transactions is broadcasted to the website when the website is identified.

[0020] Certain embodiments disclosed herein include the method, non-transitory computer readable medium, or system noted above or below, further including or being configured to perform the following steps: proactively identifying the website by performing web crawling at least a portion of the Internet.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] The subject matter disclosed herein is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the disclosed embodiments will be apparent from the following detailed description taken in conjunction with the accompanying drawings.

[0022] Figures 1A-B are network diagrams utilized to describe various disclosed embodiments.

[0023] Figure 2 is a logical diagram of a module according to an embodiment.

[0024] Figure 3 is a flowchart illustrating a method for securing decentralized applications according to an embodiment.

[0025] Figure 4 is a flowchart illustrating a method for discovering websites according to an embodiment.

[0026] Figure 5 is a schematic diagram of a web scanner according to an embodiment.

DETAILED DESCRIPTION

[0027] The various disclosed embodiments include techniques for securing decentralized applications (dApps). More specifically, the disclosed embodiments include discovery and scanning techniques for discovering websites, identifying websites being implemented as dApps, and determining whether dApps are malicious.

[0028] In an embodiment, tests are performed by extracting calls from dApps and analyzing the extracted calls for potentially malicious behavior. To this end, calls made by a dApp are extracted by impersonating wallet addresses. More specifically, transactions are generated based on sample transactions from one or more known wallet addresses, and the generated transactions are broadcast to websites. Calls such as remote procedure calls (RPCs) are extracted from responses to those transactions. The extracted calls are analyzed in order to determine whether the behavior represented in those extracted calls is anomalous. Calls demonstrating anomalous behavior (e.g., based on requested activities, order of calls, timing of calls, etc.) may be determined to be malicious or potentially malicious, and websites which transmitted malicious or otherwise anomalous calls may be flagged as malicious or potentially malicious.

[0029] In an embodiment, the testing may be performed on one or more websites found during a discovery process. The discovery process may be performed in order to identify websites which may need to be scanned for potentially malicious behavior. In some embodiments, the discovery process utilizes data indicating websites available via the Internet which may be indicative of potentially new websites in order to identify websites that exist on the Internet without requiring crawling the Internet to identify those websites. Once the websites have been identified, transactions may be generated and sent to those websites, and any responses received from those websites may be analyzed in order to determine whether the websites are potentially malicious dApps.

[0030] In some embodiments, a module may be deployed and integrated with a wallet so as to intercept requests such as wallet calls to the wallet, thereby ensuring that requests from the website are not allowed to reach the wallet, at least until the website is cleared as not being malicious. In a further embodiment, the module has limited communication capabilities that prevent leaking of data from the wallet with which the module is integrated while allowing for testing websites with which the module communicates. For example,

the module may be configured to transmit only transactions and simulated RPC responses, and not to transmit any other data of the wallet or of a device associated with a user of the wallet.

[0031] In accordance with various embodiments, the module may be realized as a binary large object (blob) pushed to the device or system on which the module is deployed, for example via a content delivery network. Updated blobs including updated blob data and/or functions may be pushed to the device or system periodically, thereby allowing for updating the data used for testing, for example as new websites are discovered by a system managing the modules.

[0032] In this regard, it has been identified that some attackers who seek to exploit decentralized applications attempt to determine the identity of users in order to determine how those users might be exploited. To this end, such attackers may attempt to intercept wallet calls as part of the attempt to identify users. Websites may engage in malicious behavior in different ways, but many malicious web 3.0 websites realized as dApps involve exploiting wallet calls. Accordingly, it has been further identified that potentially malicious behavior of dApps can be detected by analyzing wallet calls. The disclosed embodiments provide techniques for analyzing such wallet calls as well as techniques providing new ways of obtaining wallet call data for analysis.

[0033] In addition to cybersecurity benefits related to mitigating activity of malicious dApps, the disclosed embodiments provide techniques for discovering websites which may be useful in identifying websites which could be dApps. In this regard, it is noted that determining whether a website is malicious or not first requires identifying that the website exists. New websites are being created constantly, and manual indexing of all potential websites every time dApp scanning is desired would be cumbersome and require an enormous amount of computing resources. To this end, the disclosed embodiments further include various techniques for website discovery, and in particular more efficient discovery techniques that allow for identifying new websites. Moreover, by identifying new websites, the number of websites needing to be analyzed in order to determine if they are malicious dApps is reduced, thereby further conserving computing resources.

[0034] Further, the discovery techniques described herein may be utilized to identify potentially malicious websites without requiring interacting with those websites via live

web browsers. In this regard, it is noted that some existing solutions for scraping websites can utilize live browsers (i.e., web browsers being run or otherwise capable of being run on user devices) in order to probe for websites which might be dApps. However, using a live browser carries certain security risks. The disclosed embodiments allow for using a simulated browser that mimics such a live browser in order to prompt websites to generate responses. This allows for gaining access to data relevant for discovery purposes, thereby improving security of the discovery process itself.

[0035] It is also noted that one of the major benefits of blockchain-based solutions is the ability to maintain user anonymity while ensuring trust for transactions between individuals who would otherwise not trust each other. However, cybersecurity solutions that scan a particular device or wallet may ultimately defeat this privacy benefit. To this end, the disclosed embodiments further provide techniques and solutions which utilize a limited-capability module that can be integrated within a wallet in order to generate transactions without any unauthorized exposure of data from the wallet, thereby preserving privacy. Further, some disclosed embodiments include techniques for creating such a module and for managing data of the module by periodically pushing binary large objects (blobs) to the module (e.g., by a content delivery network) or pulling blobs by the module (e.g., from a content delivery network) in order to allow for the module to be updated with new website data without requiring sending communications to any backend system, thereby further ensuring privacy and improving cybersecurity by minimizing potential sources of data leakage.

[0036] FIGS. 1A-B shows example network diagrams 100A and 100B utilized to describe the various disclosed embodiments. In the example network diagram 100A, a device 120, the Internet 130, a web scanner 140, and a content delivery network (CDN) 150 communicate via one or more networks 110.

[0037] The networks 110 may be, but are not limited to, any of or a portion of a wireless, cellular or wired network, a local area network (LAN), a wide area network (WAN), a metro area network (MAN), the Internet, the worldwide web (WWW), similar networks, and any combination thereof.

[0038] The device 120 may be, but is not limited to, a personal computer, a laptop, a tablet computer, a smartphone, a wearable computing device, a server, or any other device

capable of accessing websites running as decentralized applications (dApps) pursuant to the Web 3.0 model.

[0039] To this end, the device 120 is equipped with (as shown) or is communicatively connected to (not shown) a digital wallet 125. A digital wallet, also referred to as a wallet, is a computer program or device utilized to access assets held by a user via a decentralized ledger such as a blockchain. The digital wallet 125 is configured to access one or more decentralized networks (not shown) and, in particular, to access a distributed electronic ledger such as the distributed electronic ledger 135 of the Internet 130. The digital wallet 125 may store data such as, but not limited to, data used to access distributed electronic ledgers (e.g., access credentials), data indicating assets of an account associated with the digital wallet 125 whose ownership is recorded on one or more distributed electronic ledgers (e.g., values indicating amounts of cryptocurrency or data of digital assets such as non-fungible tokens), both, and the like.

[0040] In accordance with various disclosed embodiments, the device 120 further has a module 127 installed thereon. The module 127 is configured to perform at least a portion of the disclosed embodiments related to transaction analysis and perform transaction security activities. To this end, the module 127 may be installed as an extension on the digital wallet 125 or otherwise implemented in the digital wallet 125.

[0041] In an example implementation, the module 127 may be realized at least partially as a binary large object (blob). Such a blob is a collection of binary data stored as a single entity, for example in a database management system. In such embodiments, each blob may include a collection of data on existing websites (e.g., website data created or collected by the web scanner 140), instructions for performing functions of the module (e.g., instructions which realize the logical components of the logical diagram 200 and which enable collection of data related to wallet calls as discussed herein), or both.

[0042] In some implementations, the device 120 may be a user device owned or operated by a person who may access web 3.0 websites. In such implementations, the digital wallet 125 may be a personal digital wallet of the user and may include the user's personal assets. Further, in such implementations, the module 127 may be realized as an agent installed on the device 120.

[0043] In another implementation, the device 120 may be a server or other system owned or operated by an entity who uses web 3.0 websites in order to realize or otherwise in order to provide their services. In such cases, the digital wallet 125 may be an organization digital wallet of the entity operating the device 120 and may include business assets or other assets which may be utilized for the organization's services (e.g., cryptocurrency utilized to purchase services consumed by the device 120).

[0044] The web scanner 140 is configured to perform at least some of the disclosed techniques for website discovery and dApp scanning. More specifically, the web scanner 140 is configured to perform lightweight dApp scanning in order to efficiently analyze websites to determine whether they are dApps and whether they are potentially malicious. The web scanner 140 may further be configured to perform an efficient discovery process in order to identify potential websites available over the Internet 130 without requiring crawling the entire Internet 130. In this regard, the web scanner 140 may allow for indexing the entire Internet in an efficient manner.

[0045] The Internet 130 is an interconnected network which is at least partially designed according to the web 3.0 model in which websites are realized as decentralized applications (dApps) hosted across a plurality of nodes (not separately depicted in FIG. 1) of a decentralized network (e.g., the Internet 130 or a subset thereof). To this end, the Internet 130 is implemented via one or more distributed electronic ledgers 135. Each distributed electronic ledger 135 may be, but is not limited to, a blockchain or other ledger distributed across multiple nodes.

[0046] As noted above, such decentralized applications may need to be secured against potential cyber threats. In particular, it has been identified that attackers attempting to exploit vulnerabilities in dApps may attempt to identify users by intercepting wallet calls, so malicious websites may be implemented by attackers as dApps in order to gain access to wallet calls for their malicious purposes. The disclosed embodiments provide techniques for identifying dApps among websites available via the Internet 130 as well as for scanning the identified dApps in order to detect potentially malicious behavior.

[0047] In some embodiments, the web scanner 140 is also configured to manage the module 127 as well as any other modules utilizing existing website data. To this end, web scanner may further be configured to push data (e.g., website data, computer instructions, both,

etc.) to the device 120. The web scanner 140 may receive requests to scan websites and to determine if those websites are malicious or potentially malicious. Blobs to be deployed as the module 127 and integrated with the wallet 125 may be created by the web scanner 140 using the results of such scans such as, but not limited to, the existence of discovered websites, whether each website was determined to be malicious or not, both, and the like.

[0048] The web scanner 140 may be configured to periodically, continuously, or otherwise repeatedly push new data to the device 120, for example data including newly discovered websites. To this end, in some embodiments, the web scanner 140 may be configured to create new blobs and to push new blobs to the device 120, for example periodically. The new blobs or other data from the web scanner 140 may be sent to the device 120 via one or more networks such as, but not limited to, a content delivery network (CDN) 150. As noted above, the module may include at least two kinds of memory segments, i.e., kinds of data which are utilized to execute the module: code and data. Each memory segment may be updated through differential (diff) functions, thereby minimizing the network usage needed to deploy updated data on the device 120. Such diffs may include, but are not limited to, additions of data and code, and may further include subtractions of portions of data or code which are no longer to be used with the module 127. More specifically, for such a subtraction, previously identified threats (e.g., previously detected malicious websites) may be excluded from data pushed to the device 120, thereby removing old threats from the data to be transmitted.

[0049] FIG. 1B depicts a network diagram 100B according to another example implementation in which a module configured like the module 127 of FIG. 1A is deployed externally to the device 120. To this end, in FIG. 1B, a module 167 is deployed on a hosted module server 160. The module 167 may be configured with limited communication capabilities, may be integrated with the wallet 125, and may otherwise be implemented as discussed below in order to enable the module 167 to be utilized for testing as described herein while minimizing the exposure of any data of the device 120 or the wallet 125 to external systems (i.e., systems other than the device 120 or the hosted module system 160). Moreover, the communication capabilities of the module 167 may be limited such that even the operator of the hosted module server 160 cannot access data of the wallet 125 or otherwise of the device 120, thereby ensuring privacy.

[0050] It should be noted that FIGS. 1A-B depict the Internet merely for example purposes, but that the disclosed embodiments may be equally applied to other portions of the Internet or otherwise other networks on which dApps are hosted in at least a portion of the network in order to discover websites implemented on those networks, to determine whether those websites are dApps, and to scan any identified dApps for potentially malicious behavior.

[0051] FIG. 2 is a logical diagram 200 of a module according to an embodiment. The logical diagram 200 may reflect interactions between logical components of a module such as, but not limited to, the module 127 of FIG. 1A or the module 167 of FIG. 1B. The logical components may be realized as, for example, sets of computer-executable instructions for performing discrete portions of at least some disclosed embodiments.

[0052] The logical diagram 200 illustrates a security validation engine 210, a simulation engine 220, and a remote procedure call (RPC) node provider 230. As depicted in FIG. 2, the module represented by the logical diagram 200 includes three points of entry wherein components of the logical diagram 200 communicate with an external environment. More specifically, the security validation engine 210 has a point of entry for sending and receiving data related to validating transactions, the simulation engine 220 has a point of entry for sending and receiving data related to simulating transactions, and the RPC node provider 230 has a point of entry for sending RPC node calls.

[0053] In an embodiment, the security validation engine 210 and the simulation engine 220 may each be configured with one or more exported functions defined elsewhere and pushed to a device on which the module is to be deployed. More specifically, such functions may be included in a binary large object (blob) generated by an external system and pushed to the device in order to deploy the module on the device. Additional blobs may be created and pushed to the device, for example periodically, in order to update the functions or otherwise to update the data of the module deployed on the device. Moreover, these exported functions can be designed to avoid sending any data other than their specifically defined returned data in order to prevent the exported functions from leaking data outside of the system on which the module is deployed.

[0054] Thus, in accordance with various implementations, the module is unable to send at least some information to an external server, and the only input/output operation other

than the exported functions used for simulation and validation would include a remote procedure call to a node primitive (e.g., a JSON-RPC primitive). Additionally, the module may be treated as a read-only proxy for RPC requests received by the wallet with which the module is integrated in order to allow the module to detect both malicious or otherwise anomalous calls at the point of signature as well as malicious dApp probes of wallets based on known patterns associated with bad actors prior to any prompts for a user to sign a transaction. Thus, the module optimizes privacy and cybersecurity, both when deployed on the same device as the wallet it is integrated with and when deployed on an external system (e.g., the hosted module system 160, FIG. 1B).

[0055] In an embodiment, the security validation engine 210 is configured with a validation function that is designed to receive a RPC method (e.g., a JSON-RPC method) and to return a result indicating whether the received RPC method is malicious. In some implementations, the exported validation function may further express whether the RPC method is malicious with increased granularity, for example whether the method is benign, potentially malicious, or malicious. The exported validation function may also return a notification including an explanation of the results of its validation. The validation function may be defined with respect to one or more malicious call determination rules, which in turn may be utilized to determine whether a call (and, accordingly, the website that transmitted the call) is malicious or potentially malicious. Further, the malicious call determination rules may be defined with respect to known malicious or otherwise anomalous call behavioral data, which may be pushed as part of blobs updating the module (for example, periodically) to update such data and allow for detection of new indicators of malicious calls as those new indicators are identified.

[0056] In an embodiment, the simulation engine 220 is configured with a simulate transaction function and a raw simulate transaction function. The simulate transaction function is designed to obtain a transaction and to return a full simulation of the transferred assets indicated in the transaction as well as the potential exposure of the transaction. The raw simulate transaction function may obtain a transaction and return a full simulation for the transaction like the simulate transaction function, but for a raw byte string transaction. The simulation engine 220 may be further configured to simulate opcodes of calls, to parse changes, and to determine technical information related to user experience used

for realistic simulations such as asset differentials, exposure differentials, events, and the like.

[0057] The RPC node provider 230 is configured to access one or more decentralized networks through remote procedure calls. More specifically, the RPC node provider 230 is configured to send remote procedure calls to RPC nodes of the decentralized network and to receive transaction data from those RPC nodes. An RPC node call is a call made using a lightweight communication protocol used for facilitating interactions among software applications. Remote procedure calls may be utilized to access data from a remote system, i.e., one of the RPC nodes of the decentralized network. Such RPC nodes are configured to respond to requests to a decentralized network such as a blockchain and to send transactions to requesting devices.

[0058] The RPC node provider 230, in combination with the security validation engine 210 and the simulation engine 220, may allow the module to simulate and validate transactions prior to signing. Moreover, in some embodiments, the RPC node provider 230 allows for using only RPC communication requests to a specified node provider and without requiring sending any other data to an external server. Accordingly, the RPC node provider 230 allows for preserving data privacy. That is, in such embodiments, the computer software including the functionality of the RPC node provider 230 may be implemented on a device (e.g., the device 120) and may perform simulations involving communications between the device and a decentralized network (e.g., between a device and one or more decentralized networks accessible over the Internet) without requiring that the device provide the intercepted wallet calls to an external system such as the web scanner 140. This helps to ensure data privacy and increase cybersecurity, as the data does not need to be sent for testing on another system which would serve as another possible point of failure (i.e., if the external system is hacked, the user's data can be stolen even if the user's device and the decentralized network have not been penetrated).

[0059] FIG. 3 is a flowchart 300 illustrating a method for securing decentralized applications according to an embodiment. The method of FIG. 3 describes a testing process which can be utilized to identify malicious dApps. In an embodiment, the method is performed by the web scanner 140, FIGS. 1A-B.

[0060] At optional S310, websites may be discovered. The discovery process allows for identifying the existence of websites, for example, websites available over the Internet. In this regard, in various embodiments, the discovery process may be performed in order to perform an Internet-wide scan which is capable of discovering websites across the entire Internet. An example discovery process is discussed further below with respect to FIG. 4. As noted with respect to FIG. 4, in an embodiment, data which may be indicative of the creation of new websites or which otherwise may indicate the existence of a website is analyzed without requiring crawling through all websites in order to allow for such an Internet-wide scanning without requiring crawling through the entire Internet or another network hosting websites.

[0061] In this regard, it is noted that a list of known websites may not be available or may be based on missing information (i.e., such that the list itself is incomplete). Accordingly, using the disclosed discovery process may allow for identifying more potentially malicious dApps than existing solutions, thereby further securing devices against cyber threats. Further, the discovery process may be utilized to proactively index the entire Internet in a manner where new dApps are discovered as they come online and can be scanned for potentially malicious behavior even before user data is compromised.

[0062] At S320, a website to be analyzed is identified as a decentralized application (dApp). The identified website may be, but is not limited to, a website among a list of known websites, a website discovered during the discovery process mentioned at S310, or a combination thereof. To this end, in some embodiments, S320 includes receiving a list of known websites, for example, a list of websites known to a user who wishes to have those websites analyzed for potential cybersecurity risks. In some embodiments, the website to be analyzed is a website found and determined to be a dApp during a discovery process (for example, the discovery process discussed below with respect to FIG. 4).

[0063] In an embodiment, S320 further includes determining whether one or more websites are dApps or otherwise whether the websites are enabled for Web 3.0. In a further embodiment, such a determination may include performing a static, lightweight scan (for example, as a static scraping process) that can be performed efficiently (e.g., as compared to the full testing process described below) in order to eliminate some potentially irrelevant (i.e., non-dApp) websites before testing those websites. To this end,

if a website scanned in this manner is not determined to be a dApp, the website may not be tested in subsequent steps.

[0064] In an embodiment, S320 may further include determining whether the website is a new website to be tested for potentially malicious behavior. More specifically, the discovery process may yield discovery of newly created websites such as, but not limited to, websites created since a last iteration of discovery, of scanning, or both. The websites identified as dApps to be tested may include all websites found during discovery, or may only include newly created websites found during discovery. In this regard, websites which have already been discovered and scanned may be excluded from the discovery process.

[0065] At S330, calls made by the website identified at S320 are extracted for analysis. In an embodiment, S330 including generating or causing generation of mock transactions that imitate transactions which would have been generated by a particular digital wallet (e.g., a wallet belonging to a particular user or entity). When those mock transactions are transmitted to a website, the responses to those mock transactions may include requests from the website, such as requests in the form of calls. Such requests may include, but are not limited to, network and remote procedure calls (e.g., JSON-RPC calls). Such calls may further include wallet calls made to a wallet deployed on a device communicating with the website.

[0066] The mock transactions used to trigger calls from the website may be created so as to impersonate one or more wallet addresses (e.g., wallet addresses owned or operated by one or more particular entities). Such impersonation may cause the website to determine that the mock transactions are being transmitted by valuable targets and, thus, to determine that calls should be sent to those targets. To this end, in an embodiment, one or more sample transactions from a particular wallet may be processed and utilized to generate a mock transaction that imitates the transactions which would be sent by the wallet. In response to the transmission of the mock transactions, potentially malicious dApps may respond with calls. As noted above, some of those calls may be utilized in an attempt to gain improper access to user data.

[0067] At S340, the extracted calls are analyzed. The extracted calls may include, but are not limited to, one or more requests from the website received responsive to the transaction.

More specifically, the requests may include one or more wallet calls to the integrated wallet. As noted above, such wallet calls may be made by malicious sites in order to attempt to access data indicating information related to the user's identity or otherwise to obtain unauthorized access to user data via the wallet.

[0068] In this regard, it is noted that many malicious dApps behave differently when used by different wallets. Accordingly, creating mock transactions as discussed at S340 and analyzing the response from the website allows for effectively tricking potential dApps into determining that they are interacting with high value wallet addresses, which in turn triggers malicious dApps to generate malicious responses intended to obtain undesirable access to user data. Transaction-related data included in such responses, such as RPCs, may be analyzed and compared against a database of malicious call identifiers in order to determine whether a call and, consequently, the website that transmitted the call, is malicious.

[0069] At S350, it is determined whether the website is malicious based on the analysis of the extracted calls. In an embodiment, it is determined whether the calls included among the responses from the website demonstrate anomalous behavior (e.g., based on requested activities, order of calls, timing of calls, etc.) and, if so, the website is determined to be malicious. In other words, in some embodiments, if a call transmitted by a website implemented as a dApp is flagged as malicious, the website (and, therefore, the entire dApp) may be flagged as malicious.

[0070] In an embodiment, S350 includes performing one or more checks for malicious dApps according to one or more predetermined malicious call detection rules. The malicious call detection rules may include rules for determining whether a call is malicious such that a dApp may be determined as malicious when it sends response data including one or more malicious calls. In some embodiments, the malicious call detection rules may define criteria related to anomalous call behavior such that calls demonstrating anomalous call behavior (either individually or in combination with other calls) may be identified as malicious calls.

[0071] More specifically, the malicious dApp detection rules may be defined with respect to historical data related to call behavior such as, but not limited to, activities or other information indicated in those calls, call order (i.e., an order in which calls are transmitted),

call timing, combinations thereof, and the like. As a non-limiting example, historical on-chain activity data indicating on-chain activities that occurred on a blockchain during prior communications between the device and historical websites may be compared to on-chain activities indicated in the response from the website being tested in order to determine whether the behavior is unusual as compared to the historical data per the malicious call detection rules.

[0072] In this regard, it is noted that a blockchain is state-based, such that having an address of a website on a blockchain can allow for comparing a current state of the blockchain with respect to the website to previous states of the blockchain. More specifically, an address of a website being tested can be utilized in order to determine if the address was previously indexed in the pipeline of the blockchain. Accordingly, the address information can serve as a baseline from which anomalies (which may indicate malicious behavior) can be detected in order to determine whether a transaction is likely malicious.

[0073] In some embodiments, S350 further includes storing results of the malicious dApp determination in a database or otherwise building a security engine based on the results. Such a database may include data related to previous website determinations, and may be utilized in order to determine whether a call made by a website is coming from a malicious source. When a module is deployed to a device or system (for example, as described below with respect to S370), the data included in the module may include data on any previously scanned websites. In this regard, the modules deployed in this manner may act as a compressed version of a security engine with similar capabilities in terms of functionality and a subset of the data that can be updated to bring the module's data in line with that of the security engine (e.g., as more websites are scanned, results of those scans may be used to update the data of the security engine, and when new modules are deployed, the data of those new modules may include results of recent scans).

[0074] At optional S360, one or more mitigation actions may be performed with respect to the website. The mitigation actions may include, but are not limited to, blacklisting the website, blocking certain types of communications (e.g., wallet calls) between a device and the website, blocking all communications between the device and the website, severing a connection between the device and the website, generating a notification indicating that the website is malicious, combinations thereof, and the like.

[0075] At optional S370, a module may be deployed based on results of one or more scans including the scan performed at S330 through S350. In an embodiment, the module is deployed in a system (e.g., the device 120 of FIG. 1A or the hosted module system 160 of FIG. 1B). To this end, the module deployed at S330 may be the module 127 of FIG. 1A or the module 167 of FIG. 1B. Further, the module may include the logical components depicted with respect to FIG. 2. In a further embodiment, the module has limited capabilities, i.e., the module is only configured to perform certain kinds of communications and is incapable of performing other kinds of communications.

[0076] In an embodiment, deploying the module at least includes creating module data for the module and transmitting the module to a destination (e.g., the device 120 or the hosted module server 160, FIGS. 1A-B). The module data may include, but is not limited to, data indicating whether dApps are malicious determined as described above with respect to S350, computer-readable instructions for executing one or more functions of the module, both, and the like. When the module is an update to an existing module, the created module data may include only differentials indicating changes from the existing module instead of sending the total amount of data of the entire module.

[0077] In yet a further embodiment, such limited communication capabilities may include using remote procedure calls (RPCs) as the only input/output (I/O) operation that the module is configured to send to websites such that the module cannot send any other information to an external server. Such limited capabilities can be utilized to prevent the module from sending any data related to the user outside of the device in which the module is deployed. This ensures privacy and prevents unauthorized exposure of data of the device while still allowing the module to communicate with potentially malicious websites in a manner that enables testing transactions via the module.

[0078] As noted above, the module that is deployed may be realized as a binary large object (blob) pushed to the device or system on which the module is to be deployed. Also noted above, such a blob may include differential data indicating differences from a prior version of a blob in order to reduce the amount of network resources needed to transmit the blob. To this end, S370 may include transmitting such a blob, for example via a content delivery network, where the blob is used to create a new module instance (e.g., a new module or

an update to an existing module) at a destination (e.g., the device 120 or the hosted module server 160).

[0079] In an embodiment, the module may be integrated with a digital wallet of a device (e.g., the wallet 125 of the device 120, FIGS. 1A-B). The integration of the module is performed such that the module is configured to intercept calls to the wallet once integrated. To this end, in an embodiment, the wallet integration includes adapting the module to intercept calls to the wallet. As noted above, the module is configured to simulate and validate transactions. By intercepting calls to the wallet, the module may imitate transactions by the wallet while preventing the responses to those transactions from being transmitted to the wallet, thereby allowing for analyzing transactions for potentially malicious behavior without exposing the wallet to calls from potentially malicious websites prior to such analysis. The code for integrating the module with the wallet may be included in any blobs or other data transmitted to the device or system on which the module is to be deployed such that, when that code is executed, the module integrates with the wallet.

[0080] In an embodiment, execution continues with S310, where another website from a list of known websites or another website found during discovery is identified as the next website to be analyzed. It should be noted that FIG. 3 depicts an embodiment in which such testing is performed sequentially over iterations, but that the testing of any or all of the websites to determine whether they are malicious may be performed in parallel without departing from the scope of the disclosure.

[0081] FIG. 4 is a flowchart 400 illustrating a method for discovering websites according to an embodiment. In an embodiment, the method is performed by the module 127, FIG. 1.

[0082] At optional S410, a list of website locations for known websites may be received. Such a list may be received, for example, in order to establish known websites as contrasted with new or otherwise unknown websites. As noted herein, known existing websites may be ignored during discovery in at least some implementations. As a non-limiting example where the discovery is being used as part of a process for identifying malicious dApps (e.g., as discussed above with respect to FIG. 3), websites which were previously found during discovery and tested for potentially malicious behavior may be ignored during subsequent discovery iterations and not tested again, i.e., assuming that the dApp status remains the same and that the website is

[0083] At S420, data which may indicate locations of websites is obtained. In particular, such data indicates information related to newly created websites. Such data may include, but is not limited to, website security certificates, domain name system (DNS) data, advertisement protocol data, combinations thereof, and the like. Such data indicates websites on a network such as the Internet. In some embodiments, S420 may further include performing web crawling in order to identify at least some of the websites. In such embodiments, new websites may be proactively detected. To this end, in a further embodiment, proactively detecting the new websites may include subscribing to systems on the Internet that provide indications of creation of new websites.

[0084] In this regard, it is noted that data such as website security certificates, DNS data, and advertisement protocol data are kinds of data which indicate known websites. Accordingly, analyzing this data allows for identifying potential websites without requiring crawling in order to find the websites. This allows for performing the discovery process much more efficiently than processes which involve crawling the entire Internet.

[0085] At S430, websites are identified based on the obtained data. In an embodiment, S430 includes analyzing the obtained data for identifiers of websites. In some embodiments, S430 may further include indexing the identified websites. Such indexing may be performed, for example, in order to build an index of the entire Internet over time.

[0086] FIG. 5 is an example schematic diagram of the web scanner 140 according to an embodiment. The web scanner 140 includes a processing circuitry 510 coupled to a memory 520, a storage 530, and a network interface 540. In an embodiment, the components of the web scanner 140 may be communicatively connected via a bus 550.

[0087] The processing circuitry 510 may be realized as one or more hardware logic components and circuits. For example, and without limitation, illustrative types of hardware logic components that can be used include field programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), Application-specific standard products (ASSPs), system-on-a-chip systems (SOCs), graphics processing units (GPUs), tensor processing units (TPUs), general-purpose microprocessors, microcontrollers, digital signal processors (DSPs), and the like, or any other hardware logic components that can perform calculations or other manipulations of information.

[0088] The memory 520 may be volatile (e.g., random access memory, etc.), non-volatile (e.g., read only memory, flash memory, etc.), or a combination thereof.

[0089] In one configuration, software for implementing one or more embodiments disclosed herein may be stored in the storage 530. In another configuration, the memory 520 is configured to store such software. Software shall be construed broadly to mean any type of instructions, whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise. Instructions may include code (e.g., in source code format, binary code format, executable code format, or any other suitable format of code). The instructions, when executed by the processing circuitry 510, cause the processing circuitry 510 to perform the various processes described herein.

[0090] The storage 530 may be magnetic storage, optical storage, and the like, and may be realized, for example, as flash memory or other memory technology, compact disk- read only memory (CD-ROM), Digital Versatile Disks (DVDs), or any other medium which can be used to store the desired information.

[0091] The network interface 540 allows the web scanner 140 to communicate with, for example, digital wallet 125, the decentralized network 135, the CDN 150, the hosted module server 160, a combination thereof, and the like.

[0092] It should be understood that the embodiments described herein are not limited to the specific architecture illustrated in FIG. 5, and other architectures may be equally used without departing from the scope of the disclosed embodiments.

[0093] It is important to note that the embodiments disclosed herein are only examples of the many advantageous uses of the innovative teachings herein. In general, statements made in the specification of the present application do not necessarily limit any of the various claimed embodiments. Moreover, some statements may apply to some inventive features but not to others. In general, unless otherwise indicated, singular elements may be in plural and vice versa with no loss of generality. In the drawings, like numerals refer to like parts through several views.

[0094] The various embodiments disclosed herein can be implemented as hardware, firmware, software, or any combination thereof. Moreover, the software may be implemented as an application program tangibly embodied on a program storage unit or computer readable medium consisting of parts, or of certain devices and/or a combination

of devices. The application program may be uploaded to, and executed by, a machine comprising any suitable architecture. Preferably, the machine is implemented on a computer platform having hardware such as one or more central processing units ("CPUs"), a memory, and input/output interfaces. The computer platform may also include an operating system and microinstruction code. The various processes and functions described herein may be either part of the microinstruction code or part of the application program, or any combination thereof, which may be executed by a CPU, whether or not such a computer or processor is explicitly shown. In addition, various other peripheral units may be connected to the computer platform such as an additional data storage unit and a printing unit. Furthermore, a non-transitory computer readable medium is any computer readable medium except for a transitory propagating signal.

[0095] All examples and conditional language recited herein are intended for pedagogical purposes to aid the reader in understanding the principles of the disclosed embodiment and the concepts contributed by the inventor to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the disclosed embodiments, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure.

[0096] It should be understood that any reference to an element herein using a designation such as "first," "second," and so forth does not generally limit the quantity or order of those elements. Rather, these designations are generally used herein as a convenient method of distinguishing between two or more elements or instances of an element. Thus, a reference to first and second elements does not mean that only two elements may be employed there or that the first element must precede the second element in some manner. Also, unless stated otherwise, a set of elements comprises one or more elements.

[0097] As used herein, the phrase "at least one of" followed by a listing of items means that any of the listed items can be utilized individually, or any combination of two or more of

the listed items can be utilized. For example, if a system is described as including “at least one of A, B, and C,” the system can include A alone; B alone; C alone; 2A; 2B; 2C; 3A; A and B in combination; B and C in combination; A and C in combination; A, B, and C in combination; 2A and C in combination; A, 3B, and 2C in combination; and the like.

CLAIMS

What is claimed is:

1. A method for decentralized application scanning, comprising:
 - generating a plurality of first transactions based on a plurality of second transactions, wherein the plurality of second transactions is from a digital wallet;
 - broadcasting the generated plurality of transactions to a website, wherein the website returns a plurality of responses to the plurality of first transactions;
 - extracting a plurality of calls from the plurality of responses returned by the website; and
 - determining whether each extracted call is an anomalous call by analyzing the extracted plurality of calls.
2. The method of claim 1, wherein determining whether each extracted call is an anomalous call results in determining at least one anomalous call among the plurality of extracted calls, further comprising:
 - performing at least one mitigation action with respect to the at least one anomalous call.
3. The method of claim 2, further comprising:
 - identifying the website as a malicious website based on the determined at least one anomalous call, wherein the at least one mitigation action is performed when the website is identified as a malicious website.
4. The method of claim 1, wherein the digital wallet is a first digital wallet, further comprising:
 - deploying a module to a second digital wallet, wherein the module is configured to intercept requests to the second digital wallet, wherein the plurality of first transactions is broadcast via the second digital wallet, wherein the plurality of responses from which the plurality of calls is extracted is received by the second digital wallet.

5. The method of claim 4, wherein the module is configured to transmit only a subset of types of transactions and simulated responses from among a set of potential types of transactions and simulated responses.
6. The method of claim 5, wherein the subset of types of transactions and simulated responses includes remote procedure calls as the only input/output operation which the module is configured to send to websites.
7. The method of claim 4, wherein the module is a binary large object, further comprising:
 - pushing an updated version of the binary large object to the second wallet, wherein the updated version of the binary large object includes differential data indicating differences from a prior version of the binary large object, wherein the plurality of transactions is broadcast via the second digital wallet using the updated version of the binary large object.
8. The method of claim 1, wherein determining whether each extracted call is an anomalous call further comprises:
 - determining whether each extracted call requests an anomalous transaction, wherein each extracted call which requests an anomalous transaction is determined to be an anomalous call.
9. The method of claim 1, further comprising:
 - identifying the website based on data indicating locations of newly created websites, wherein the plurality of transactions is broadcasted to the website when the website is identified.
10. The method of claim 1, further comprising:
 - proactively identifying the website by performing web crawling at least a portion of the Internet.

11. A non-transitory computer readable medium having stored thereon instructions for causing a processing circuitry to execute a process, the process comprising:

generating a plurality of first transactions based on a plurality of second transactions, wherein the plurality of second transactions is from a digital wallet;

broadcasting the generated plurality of transactions to a website, wherein the website returns a plurality of responses to the plurality of first transactions;

extracting a plurality of calls from the plurality of responses returned by the website; and

determining whether each extracted call is an anomalous call by analyzing the extracted plurality of calls.

12. A system for decentralized application scanning, comprising:

a processing circuitry; and

a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to:

generate a plurality of first transactions based on a plurality of second transactions, wherein the plurality of second transactions is from a digital wallet;

broadcast the generated plurality of transactions to a website, wherein the website returns a plurality of responses to the plurality of first transactions;

extract a plurality of calls from the plurality of responses returned by the website; and

determine whether each extracted call is an anomalous call by analyzing the extracted plurality of calls.

13. The system of claim 12, wherein determining whether each extracted call is an anomalous call results in determining at least one anomalous call among the plurality of extracted calls, wherein the system is further configured to:

perform at least one mitigation action with respect to the at least one anomalous call.

14. The system of claim 13, wherein the system is further configured to:

identify the website as a malicious website based on the determined at least one anomalous call, wherein the at least one mitigation action is performed when the website is identified as a malicious website.

15. The system of claim 12, wherein the system is further configured to:

deploy a module to a second digital wallet, wherein the module is configured to intercept requests to the second digital wallet, wherein the plurality of first transactions is broadcast via the second digital wallet, wherein the plurality of responses from which the plurality of calls is extracted is received by the second digital wallet.

16. The system of claim 15, wherein the module is configured to transmit only a subset of types of transactions and simulated responses from among a set of potential types of transactions and simulated responses.

17. The system of claim 16, wherein the subset of types of transactions and simulated responses includes remote procedure calls as the only input/output operation which the module is configured to send to websites.

18. The system of claim 15, wherein the module is a binary large object, wherein the system is further configured to:

push an updated version of the binary large object to the second wallet, wherein the updated version of the binary large object includes differential data indicating differences from a prior version of the binary large object, wherein the plurality of transactions is broadcast via the second digital wallet using the updated version of the binary large object.

19. The system of claim 12, wherein the system is further configured to:

determine whether each extracted call requests an anomalous transaction, wherein each extracted call which requests an anomalous transaction is determined to be an anomalous call.

20. The system of claim 12, wherein the system is further configured to:
identify the website based on data indicating locations of newly created websites,
wherein the plurality of transactions is broadcasted to the website when the website is
identified.
21. The system of claim 12, wherein the system is further configured to:
proactively identify the website by performing web crawling at least a portion of the
Internet.