

METHOD AND SYSTEM FOR IMPROVED MALWARE DETECTION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority from U.S. Application No. 16/400,087, filed May 1, 2019, which claims the benefit of U.S. Provisional Patent Application Serial No. 62/676,559, filed May 25, 2018, the contents of each of which are hereby incorporated by reference in their respective entireties.

BACKGROUND

1. Field of the Disclosure

[0002] This technology generally relates to methods and systems for detecting malware, and more particularly, to methods and systems for real-time comparison of expected web content versus executed and rendered web content to detect malware that is present on a client device.

2. Background Information

[0003] There are increasing numbers of attacks directed toward various entities, including individual persons, financial services organizations and networks, and other commercial entities, by malicious actors who target end user devices by utilizing malware to hijack user sessions or surreptitiously obtain confidential information, such as identification information that relates to individual persons or financial accounts. In some instances, web pages are modified by using webinject malware so as to change the client-side web page execution in order to render a fraudulent web page in a web browser of an unsuspecting user. As a result, an unsuspecting user that engages with malicious content of the modified web page may unknowingly provide confidential information to third parties.

[0004] Although conventional computer systems may have installed commercially available software, such as anti-virus software, that is designed to detect computer viruses, such software is intended for general use, and as such, is not tailored for any specific type of web page. As a result, such software may be ineffective. Therefore, there is a need for a tool that is designed to detect malware with respect to specific web pages.

SUMMARY

[0005] The present disclosure, through one or more of its various aspects, embodiments, and/or specific features or sub-components, provides, inter alia, various systems, servers, devices, methods, media, programs, and platforms for injecting client-side code into web pages prior to execution thereof in order to detect malware. The various aspects, embodiments, features, and/or sub-components provide optimized processes of injecting client-side code into web pages prior to execution thereof in order to detect malware

[0006] According to an aspect of the present disclosure, a method for detecting malware is provided. The method is implemented by a processor on a computing device. The method includes: receiving, by the processor, a request for a first web page; determining, by the processor, at least one expected attribute of the first web page; generating, by the processor, a modified web page by combining the first web page with a predetermined set of code, the predetermined set of code being configured to detect whether the malware is present on a client device; transmitting, to the client device, the modified web page; receiving, from the client device, an executed version of the modified web page; determining, by the processor, at least one attribute of the executed version of the modified web page; comparing, by the processor, the at least one expected attribute of the first web page with the at least one attribute of the executed version of the modified web page; and determining, by the processor, whether the malware is present on the client device based on a result of the comparing.

[0007] The method may further include transmitting, to the client device, an alert message indicating that the malware is present on the client device.

[0008] The at least one expected attribute of the first web page may include a first number of JavaScript elements that are associated with the first web page. The determination of whether the malware is present on the client device may further include comparing the first number of JavaScript elements with a second number of JavaScript elements that are associated with the executed version of the modified web page.

[0009] The at least one expected attribute of the first web page may include a first set of identifiers that correspond to JavaScript elements that are associated with the first web page. The determination of whether the malware is present on the client device may further include comparing the first set of identifiers with a second set of identifiers that correspond to JavaScript elements that are associated with the executed version of the modified web page.

[0010] The at least one expected attribute of the first web page may include a first set of sources of JavaScript elements that are associated with the first web page. The determination of

whether the malware is present on the client device may further include comparing the first set of sources with a second set of sources of JavaScript elements that are associated with the executed version of the modified web page.

[0011] The at least one expected attribute of the first web page may include a first set of Uniform Resource Identifiers (URIs) that are associated with the first web page. The determination of whether the malware is present on the client device may further include comparing the first set of URIs with a second set of URIs that are associated with the executed version of the modified web page.

[0012] The at least one expected attribute of the first web page may include a first set of JavaScript function names that are associated with the first web page. The determination of whether the malware is present on the client device may further include comparing the first set of JavaScript function names with a second set of JavaScript function names that are associated with the executed version of the modified web page.

[0013] The at least one expected attribute of the first web page may include first information that relates to JavaScript event handlers that are associated with the first web page. The determination of whether the malware is present on the client device may further include comparing the first information with second information that relates to JavaScript event handlers that are associated with the executed version of the modified web page.

[0014] The at least one expected attribute of the first web page may include a first set of cookies that are associated with the first web page. The determination of whether the malware is present on the client device may further include comparing the first set of cookies with a second set of cookies that are associated with the executed version of the modified web page.

[0015] The method may further include: when a determination is made that malware is not present on the client device, storing information that relates to the first web page in a first database that stores a first list of requested and unmodified web pages and content; and when a determination is made that malware is present on the client device, storing information that relates to the first web page in a second database that stores a second list of requested and modified web pages and content.

[0016] According to another aspect of the present disclosure, a computing device configured to implement an execution of a method for detecting malware is provided. The computing device includes a display screen, a processor, a memory, and a communication interface coupled to each of the processor, the memory, and the display screen. When the method is being executed, the processor is configured to: receive, via the communication interface, a

request for a first web page; determine at least one expected attribute of the first web page; generate a modified web page by combining the first web page with a predetermined set of code, the predetermined set of code being configured to detect whether the malware is present on a client device; transmit, to the client device via the communication interface, the modified web page; receive, from the client device via the communication interface, an executed version of the modified web page; determine at least one attribute of the executed version of the modified web page; compare the at least one expected attribute of the first web page with the at least one attribute of the executed version of the modified web page; and determine whether the malware is present on the client device based on a result of the comparison.

[0017] The processor may be further configured to transmit, to the client device via the communication interface, an alert message indicating that the malware is present on the client device.

[0018] The at least one expected attribute of the first web page may include a first number of JavaScript elements that are associated with the first web page. The processor may be further configured to determine whether the malware is present on the client device by comparing the first number of JavaScript elements with a second number of JavaScript elements that are associated with the executed version of the modified web page.

[0019] The at least one expected attribute of the first web page may include a first set of identifiers that correspond to JavaScript elements that are associated with the first web page. The processor may be further configured to determine whether the malware is present on the client device by comparing the first set of identifiers with a second set of identifiers that correspond to JavaScript elements that are associated with the executed version of the modified web page.

[0020] The at least one expected attribute of the first web page may include a first set of sources of JavaScript elements that are associated with the first web page. The processor may be further configured to determine whether the malware is present on the client device by comparing the first set of sources with a second set of sources of JavaScript elements that are associated with the executed version of the modified web page.

[0021] The at least one expected attribute of the first web page may include a first set of Uniform Resource Identifiers (URIs) that are associated with the first web page. The processor may be further configured to determine whether the malware is present on the client device by comparing the first set of URIs with a second set of URIs that are associated with the executed version of the modified web page.

[0022] The at least one expected attribute of the first web page may include a first set of JavaScript function names that are associated with the first web page. The processor may be further configured to determine whether the malware is present on the client device by comparing the first set of JavaScript function names with a second set of JavaScript function names that are associated with the executed version of the modified web page.

[0023] The at least one expected attribute of the first web page may include first information that relates to JavaScript event handlers that are associated with the first web page. The processor may be further configured to determine whether the malware is present on the client device by comparing the first information with second information that relates to JavaScript event handlers that are associated with the executed version of the modified web page.

[0024] The at least one expected attribute of the first web page may include a first set of cookies that are associated with the first web page. The processor may be further configured to determine whether the malware is present on the client device by comparing the first set of cookies with a second set of cookies that are associated with the executed version of the modified web page.

[0025] When a determination is made that malware is not present on the client device, the processor may be further configured to store, in the memory, information that relates to the first web page in a first database that stores a first list of safe web pages. When a determination is made that malware is present on the client device, the processor may be further configured to store, in the memory, information that relates to the first web page in a second database that stores a second list of unsafe web pages.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] The present disclosure is further described in the detailed description which follows, in reference to the noted plurality of drawings, by way of non-limiting examples of preferred embodiments of the present disclosure, in which like characters represent like elements throughout the several views of the drawings.

[0027] FIG. 1 illustrates an exemplary computer system for implementing a method for detecting malware.

[0028] FIG. 2 illustrates an exemplary diagram of a network environment.

[0029] FIG. 3 shows an exemplary system for implementing a method for detecting malware.

[0030] FIG. 4 is a flowchart of an exemplary process for implementing a method for detecting malware.

DETAILED DESCRIPTION

[0031] Through one or more of its various aspects, embodiments and/or specific features or sub-components of the present disclosure, are intended to bring out one or more of the advantages as specifically described above and noted below.

[0032] The examples may also be embodied as one or more non-transitory computer readable media having instructions stored thereon for one or more aspects of the present technology as described and illustrated by way of the examples herein. The instructions in some examples include executable code that, when executed by one or more processors, cause the processors to carry out steps necessary to implement the methods of the examples of this technology that are described and illustrated herein.

[0033] FIG. 1 is an exemplary system for use in accordance with the embodiments described herein. The system 100 is generally shown and may include a computer system 102, which is generally indicated.

[0034] The computer system 102 may include a set of instructions that can be executed to cause the computer system 102 to perform any one or more of the methods or computer based functions disclosed herein, either alone or in combination with the other described devices. The computer system 102 may operate as a standalone device or may be connected to other systems or peripheral devices. For example, the computer system 102 may include, or be included within, any one or more computers, servers, systems, communication networks or cloud environment. Even further, the instructions may be operative in such cloud-based computing environment.

[0035] In a networked deployment, the computer system 102 may operate in the capacity of a server or as a client user computer in a server-client user network environment, a client user computer in a cloud computing environment, or as a peer computer system in a peer-to-peer (or distributed) network environment. The computer system 102, or portions thereof, may be implemented as, or incorporated into, various devices, such as a personal computer, a tablet computer, a set-top box, a personal digital assistant, a mobile device, a palmtop computer, a laptop computer, a desktop computer, a communications device, a wireless smart phone, a personal trusted device, a wearable device, a global positioning satellite (GPS) device, a web appliance, or any other machine capable of executing a set of instructions (sequential or

otherwise) that specify actions to be taken by that machine. Further, while a single computer system 102 is illustrated, additional embodiments may include any collection of systems or sub-systems that individually or jointly execute instructions or perform functions. The term “system” shall be taken throughout the present disclosure to include any collection of systems or sub-systems that individually or jointly execute a set, or multiple sets, of instructions to perform one or more computer functions.

[0036] As illustrated in FIG. 1, the computer system 102 may include at least one processor 104. The processor 104 is tangible and non-transitory. As used herein, the term “non-transitory” is to be interpreted not as an eternal characteristic of a state, but as a characteristic of a state that will last for a period of time. The term “non-transitory” specifically disavows fleeting characteristics such as characteristics of a particular carrier wave or signal or other forms that exist only transitorily in any place at any time. The processor 104 is an article of manufacture and/or a machine component. The processor 104 is configured to execute software instructions in order to perform functions as described in the various embodiments herein. The processor 104 may be a general purpose processor or may be part of an application specific integrated circuit (ASIC). The processor 104 may also be a microprocessor, a microcomputer, a processor chip, a controller, a microcontroller, a digital signal processor (DSP), a state machine, or a programmable logic device. The processor 104 may also be a logical circuit, including a programmable gate array (PGA) such as a field programmable gate array (FPGA), or another type of circuit that includes discrete gate and/or transistor logic. The processor 104 may be a central processing unit (CPU), a graphics processing unit (GPU), or both. Additionally, any processor described herein may include multiple processors, parallel processors, or both. Multiple processors may be included in, or coupled to, a single device or multiple devices.

[0037] The computer system 102 may also include a computer memory 106. The computer memory 106 may include a static memory, a dynamic memory, or both in communication. Memories described herein are tangible storage mediums that can store data and executable instructions, and are non-transitory during the time instructions are stored therein. Again, as used herein, the term “non-transitory” is to be interpreted not as an eternal characteristic of a state, but as a characteristic of a state that will last for a period of time. The term “non-transitory” specifically disavows fleeting characteristics such as characteristics of a particular carrier wave or signal or other forms that exist only transitorily in any place at any time. The memories are an article of manufacture and/or machine component. Memories described herein are computer-readable mediums from which data and executable instructions can be read by a computer. Memories as described herein may be random access memory

(RAM), read only memory (ROM), flash memory, electrically programmable read only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), registers, a hard disk, a cache, a removable disk, tape, compact disk read only memory (CD-ROM), digital versatile disk (DVD), floppy disk, blu-ray disk, or any other form of storage medium known in the art. Memories may be volatile or non-volatile, secure and/or encrypted, unsecure and/or unencrypted. Of course, the computer memory 106 may comprise any combination of memories or a single storage.

[0038] The computer system 102 may further include a display 108, such as a liquid crystal display (LCD), an organic light emitting diode (OLED), a flat panel display, a solid state display, a cathode ray tube (CRT), a plasma display, or any other type of display, examples of which are well known to skilled persons.

[0039] The computer system 102 may also include at least one input device 110, such as a keyboard, a touch-sensitive input screen or pad, a speech input, a mouse, a remote control device having a wireless keypad, a microphone coupled to a speech recognition engine, a camera such as a video camera or still camera, a cursor control device, a global positioning system (GPS) device, an altimeter, a gyroscope, an accelerometer, a proximity sensor, or any combination thereof. Those skilled in the art appreciate that various embodiments of the computer system 102 may include multiple input devices 110. Moreover, those skilled in the art further appreciate that the above-listed, exemplary input devices 110 are not meant to be exhaustive and that the computer system 102 may include any additional, or alternative, input devices 110.

[0040] The computer system 102 may also include a medium reader 112 which is configured to read any one or more sets of instructions, e.g. software, from any of the memories described herein. The instructions, when executed by a processor, can be used to perform one or more of the methods and processes as described herein. In a particular embodiment, the instructions may reside completely, or at least partially, within the memory 106, the medium reader 112, and/or the processor 110 during execution by the computer system 102.

[0041] Furthermore, the computer system 102 may include any additional devices, components, parts, peripherals, hardware, software or any combination thereof which are commonly known and understood as being included with or within a computer system, such as, but not limited to, a network interface 114 and an output device 116. The output device 116 may be, but is not limited to, a speaker, an audio out, a video out, a remote control output, a printer, or any combination thereof.

[0042] Each of the components of the computer system 102 may be interconnected and communicate via a bus 118 or other communication link. As shown in FIG. 1, the components may each be interconnected and communicate via an internal bus. However, those skilled in the art appreciate that any of the components may also be connected via an expansion bus. Moreover, the bus 118 may enable communication via any standard or other specification commonly known and understood such as, but not limited to, peripheral component interconnect, peripheral component interconnect express, parallel advanced technology attachment, serial advanced technology attachment, etc.

[0043] The computer system 102 may be in communication with one or more additional computer devices 120 via a network 122. The network 122 may be, but is not limited to, a local area network, a wide area network, the Internet, a telephony network, a short-range network, or any other network commonly known and understood in the art. The short-range network may include, for example, Bluetooth, Zigbee, infrared, near field communication, ultraband, or any combination thereof. Those skilled in the art appreciate that additional networks 122 which are known and understood may additionally or alternatively be used and that the exemplary networks 122 are not limiting or exhaustive. Also, while the network 122 is shown in FIG. 1 as a wireless network, those skilled in the art appreciate that the network 122 may also be a wired network.

[0044] The additional computer device 120 is shown in FIG. 1 as a personal computer. However, those skilled in the art appreciate that, in alternative embodiments of the present application, the computer device 120 may be a laptop computer, a tablet PC, a personal digital assistant, a mobile device, a palmtop computer, a desktop computer, a communications device, a wireless telephone, a personal trusted device, a web appliance, a server, or any other device that is capable of executing a set of instructions, sequential or otherwise, that specify actions to be taken by that device. Of course, those skilled in the art appreciate that the above-listed devices are merely exemplary devices and that the device 120 may be any additional device or apparatus commonly known and understood in the art without departing from the scope of the present application. For example, the computer device 120 may be the same or similar to the computer system 102. Furthermore, those skilled in the art similarly understand that the device may be any combination of devices and apparatuses.

[0045] Of course, those skilled in the art appreciate that the above-listed components of the computer system 102 are merely meant to be exemplary and are not intended to be exhaustive and/or inclusive. Furthermore, the examples of the components listed above are also meant to be exemplary and similarly are not meant to be exhaustive and/or inclusive.

[0046] In accordance with various embodiments of the present disclosure, the methods described herein may be implemented using a hardware computer system that executes software programs. Further, in an exemplary, non-limited embodiment, implementations can include distributed processing, component/object distributed processing, and parallel processing. Virtual computer system processing can be constructed to implement one or more of the methods or functionality as described herein, and a processor described herein may be used to support a virtual processing environment.

[0047] As described herein, various embodiments provide optimized processes of implementing a methodology for detecting malware.

[0048] Referring to FIG. 2, a schematic of an exemplary network environment 200 for implementing a methodology for detecting malware is illustrated. In an exemplary embodiment, the testing framework is executable on any networked computer platform, such as, for example, a wireless mobile communication device, i.e., a smart phone.

[0049] The methodology for detecting malware may be implemented by an Malware Detection (MD) device 202. The MD device 202 may be the same or similar to the computer system 102 as described with respect to FIG. 1. The MD device 202 may store one or more applications that can include executable instructions that, when executed by the MD device 202, cause the MD device 202 to perform actions, such as to transmit, receive, or otherwise process network messages, for example, and to perform other actions described and illustrated below with reference to the figures. The application(s) may be implemented as modules or components of other applications. Further, the application(s) can be implemented as operating system extensions, modules, plugins, or the like.

[0050] Even further, the application(s) may be operative in a cloud-based computing environment. The application(s) may be executed within or as virtual machine(s) or virtual server(s) that may be managed in a cloud-based computing environment. Also, the application(s), and even the MD device 202 itself, may be located in virtual server(s) running in a cloud-based computing environment rather than being tied to one or more specific physical network computing devices. Also, the application(s) may be running in one or more virtual machines (VMs) executing on the MD device 202. Additionally, in one or more embodiments of this technology, virtual machine(s) running on the MD device 202 may be managed or supervised by a hypervisor.

[0051] In the network environment 200 of FIG. 2, the MD device 202 is coupled to a plurality of server devices 204(1)-204(n) that hosts a plurality of databases 206(1)-206(n), and

also to a plurality of client devices 208(1)-208(n) via communication network(s) 210. A communication interface of the MD device 202, such as the network interface 114 of the computer system 102 of FIG. 1, operatively couples and communicates between the MD device 202, the server devices 204(1)-204(n), and/or the client devices 208(1)-208(n), which are all coupled together by the communication network(s) 210, although other types and/or numbers of communication networks or systems with other types and/or numbers of connections and/or configurations to other devices and/or elements may also be used.

[0052] The communication network(s) 210 may be the same or similar to the network 122 as described with respect to FIG. 1, although the MD device 202, the server devices 204(1)-204(n), and/or the client devices 208(1)-208(n) may be coupled together via other topologies. Additionally, the network environment 200 may include other network devices such as one or more routers and/or switches, for example, which are well known in the art and thus will not be described herein. This technology provides a number of advantages including methods, non-transitory computer readable media, and MD devices that efficiently inject client-side code into web pages prior to execution thereof in order to detect malware that is present on a client device.

[0053] By way of example only, the communication network(s) 210 may include local area network(s) (LAN(s)) or wide area network(s) (WAN(s)), and can use TCP/IP over Ethernet and industry-standard protocols, although other types and/or numbers of protocols and/or communication networks may be used. The communication network(s) 210 in this example may employ any suitable interface mechanisms and network communication technologies including, for example, teletraffic in any suitable form (e.g., voice, modem, and the like), Public Switched Telephone Network (PSTNs), Ethernet-based Packet Data Networks (PDNs), combinations thereof, and the like.

[0054] The MD device 202 may be a standalone device or integrated with one or more other devices or apparatuses, such as one or more of the server devices 204(1)-204(n), for example. In one particular example, the MD device 202 may include or be hosted by one of the server devices 204(1)-204(n), and other arrangements are also possible. Moreover, one or more of the devices of the MD device 202 may be in a same or a different communication network including one or more public, private, or cloud networks, for example.

[0055] The plurality of server devices 204(1)-204(n) may be the same or similar to the computer system 102 or the computer device 120 as described with respect to FIG. 1, including any features or combination of features described with respect thereto. For example, any of the server devices 204(1)-204(n) may include, among other features, one or more processors, a

memory, and a communication interface, which are coupled together by a bus or other communication link, although other numbers and/or types of network devices may be used. The server devices 204(1)-204(n) in this example may process requests received from the MD device 202 via the communication network(s) 210 according to the HTTP-based and/or JavaScript Object Notation (JSON) protocol, for example, although other protocols may also be used.

[0056] The server devices 204(1)-204(n) may be hardware or software or may represent a system with multiple servers in a pool, which may include internal or external networks. The server devices 204(1)-204(n) hosts the databases 206(1)-206(n) that are configured to store protected web page data, analytics data, and other data that relates to the malware detection methodology.

[0057] Although the server devices 204(1)-204(n) are illustrated as single devices, one or more actions of each of the server devices 204(1)-204(n) may be distributed across one or more distinct network computing devices that together comprise one or more of the server devices 204(1)-204(n). Moreover, the server devices 204(1)-204(n) are not limited to a particular configuration. Thus, the server devices 204(1)-204(n) may contain a plurality of network computing devices that operate using a master/slave approach, whereby one of the network computing devices of the server devices 204(1)-204(n) operates to manage and/or otherwise coordinate operations of the other network computing devices.

[0058] The server devices 204(1)-204(n) may operate as a plurality of network computing devices within a cluster architecture, a peer-to peer architecture, virtual machines, or within a cloud architecture, for example. Thus, the technology disclosed herein is not to be construed as being limited to a single environment and other configurations and architectures are also envisaged.

[0059] The plurality of client devices 208(1)-208(n) may also be the same or similar to the computer system 102 or the computer device 120 as described with respect to FIG. 1, including any features or combination of features described with respect thereto. For example, the client devices 208(1)-208(n) in this example may include any type of computing device that can facilitate the execution of a method for detecting malware. Accordingly, the client devices 208(1)-208(n) may be mobile computing devices, desktop computing devices, laptop computing devices, tablet computing devices, virtual machines (including cloud-based computers), or the like, that host chat, e-mail, or voice-to-text applications, for example. In an exemplary embodiment, at least one client device 208 is a wireless mobile communication device, i.e., a smart phone.

[0060] The client devices 208(1)-208(n) may run interface applications, such as standard web browsers or standalone client applications, which may provide an interface to communicate with the MD device 202 via the communication network(s) 210 in order to communicate user requests. The client devices 208(1)-208(n) may further include, among other features, a display device, such as a display screen or touchscreen, and/or an input device, such as a keyboard, for example.

[0061] Although the exemplary network environment 200 with the MD device 202, the server devices 204(1)-204(n), the client devices 208(1)-208(n), and the communication network(s) 210 are described and illustrated herein, other types and/or numbers of systems, devices, components, and/or elements in other topologies may be used. It is to be understood that the systems of the examples described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the examples are possible, as will be appreciated by those skilled in the relevant art(s).

[0062] One or more of the devices depicted in the network environment 200, such as the MD device 202, the server devices 204(1)-204(n), or the client devices 208(1)-208(n), for example, may be configured to operate as virtual instances on the same physical machine. In other words, one or more of the MD device 202, the server devices 204(1)-204(n), or the client devices 208(1)-208(n) may operate on the same physical device rather than as separate devices communicating through communication network(s) 210. Additionally, there may be more or fewer MD devices 202, server devices 204(1)-204(n), or client devices 208(1)-208(n) than illustrated in FIG. 2.

[0063] In addition, two or more computing systems or devices may be substituted for any one of the systems or devices in any example. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also may be implemented, as desired, to increase the robustness and performance of the devices and systems of the examples. The examples may also be implemented on computer system(s) that extend across any suitable network using any suitable interface mechanisms and traffic technologies, including by way of example only teletraffic in any suitable form (e.g., voice and modem), wireless traffic networks, cellular traffic networks, Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0064] The MD device 202 is described and shown in FIG. 3 as including an malware detection module 302, although it may include other rules, policies, modules, databases, or applications, for example. As will be described below, the malware detection module 302 is

configured to inject client-side code into web pages prior to execution thereof in order to detect malware that is present on a client device in an automated, efficient, scalable, and reliable manner. Based on a comparison between expected attributes of the web page and attributes of the modified web page that are determined after execution thereof, the malware detection module 302 determines whether malware is present on a client device.

[0065] An exemplary process 300 for implementing a methodology for detecting malware by utilizing the network environment of FIG. 2 is shown as being executed in FIG. 3. Specifically, a first client device 208(1) and a second client device 208(2) are illustrated as being in communication with MD device 202. In this regard, the first client device 208(1) and the second client device 208(2) may be “clients” of the MD device 202 and are described herein as such. Nevertheless, it is to be known and understood that the first client device 208(1) and/or the second client device 208(2) need not necessarily be “clients” of the MD device 202, or any entity described in association therewith herein. Any additional or alternative relationship may exist between either or both of the first client device 208(1) and the second client device 208(2) and the MD device 202, or no relationship may exist.

[0066] Further, MD device 202 is illustrated as being able to access a protected web page repository 206(1) and an analytics database 206(2). The malware detection module 302 may be configured to access these databases for implementing a methodology for detection malware.

[0067] The first client device 208(1) may be, for example, a smart phone. Of course, the first client device 208(1) may be any additional device described herein. The second client device 208(2) may be, for example, a personal computer (PC). Of course, the second client device 208(2) may also be any additional device described herein.

[0068] The process may be executed via the communication network(s) 210, which may comprise plural networks as described above. For example, in an exemplary embodiment, either or both of the first client device 208(1) and the second client device 208(2) may communicate with the MD device 202 via broadband or cellular communication. Of course, these embodiments are merely exemplary and are not limiting or exhaustive.

[0069] Upon being started, the malware detection module 302 executes a process for injecting client-side code into web pages prior to execution thereof in order to detect malware that is present on a client device. An exemplary process for implementing a method for detecting malware is generally indicated at flowchart 400 in FIG. 4.

[0070] In the process 400 of FIG. 4, at step S402, a web page request is received from a client device 208 by the malware detection module 302. The web page request may refer to any

number of web pages. The web page may be associated with a financial service organization, such as a bank, although any other type of web page may be requested. In an exemplary embodiment, the malware detection module may access the protected web page repository 206(1) to determine whether the requested web page corresponds to a web page that is to be protected.

[0071] At step S404, the malware detection module determines a set of expected attributes of the requested web page. In an exemplary embodiment, the requested web page may have any one or more of the following attributes: a number of associated JavaScript elements; a set of respective identifiers that correspond to JavaScript elements associated with the web page; a set of respective sources that correspond to JavaScript elements associated with the web page; a set of Uniform Resource Identifiers (URIs) that correspond to the web page; a set of JavaScript function names that are associated with the web page; information that corresponds to JavaScript event handlers that are associated with the web page; and a set of cookies that are associated with the web page.

[0072] At step S406, the malware detection module generates a modified version of the requested web page by injecting client-side code into the web page. The client-side code is implemented as JavaScript, which is linked or embedded into the web page. In an exemplary embodiment, when the request encompasses a plurality of web pages, the client-side code may be embedded into particular pages, such as, for example, a login page and a landing page. Alternatively, when the request encompasses a plurality of web pages, the client-side code may be linked to or embedded in all of the pages. Then, in step S408, after the modified version of the web page has been generated, the modified web page is transmitted to the client device 208.

[0073] At step S410, after the client device has executed the modified web page, the malware detection module 302 receives the results of an executed version of the modified web page. Then, at step S412, the malware detection module 302 determines attributes of the executed version of the modified web page. Similarly as discussed above with respect to step S404, the attributes to be determined may include any one or more of the following attributes: a number of associated JavaScript elements; a set of respective identifiers that correspond to JavaScript elements associated with the web page; a set of respective sources that correspond to JavaScript elements associated with the web page; a set of Uniform Resource Identifiers (URIs) that correspond to the web page; a set of JavaScript function names that are associated with the web page; information that corresponds to JavaScript event handlers that are associated with the web page; and a set of cookies that are associated with the web page.

[0074] At step S414, the malware detection module 302 compares the attributes of the executed version of the modified web page, as determined in step S412, with the expected attributes of the web page, as determined in step S404. Then, in step S416, a determination as to whether or not malware is present on the client device 208 is made based on a result of the comparison.

[0075] When the malware detection module 302 determines that no malware is present on the client device 208, then at step S418, information that corresponds to the web page and the client device is stored in a database. However, when the malware detection module 302 determines that malware is present on the client device 208, at step S420, optionally an alert is sent to the client device 208, and at step S422, the relevant information (i.e., information that relates to the web page, the client device, and the malware found) is stored in a database.

[0076] Accordingly, with this technology, an optimized process for implementing a method for detecting malware is provided. The optimized process entails injecting client-side code into web pages prior to execution thereof in order to detect malware that is present on a client device via a comparison of expected web content versus actual web content.

[0077] Although the invention has been described with reference to several exemplary embodiments, it is understood that the words that have been used are words of description and illustration, rather than words of limitation. Changes may be made within the purview of the appended claims, as presently stated and as amended, without departing from the scope and spirit of the present disclosure in its aspects. Although the invention has been described with reference to particular means, materials and embodiments, the invention is not intended to be limited to the particulars disclosed; rather the invention extends to all functionally equivalent structures, methods, and uses such as are within the scope of the appended claims.

[0078] For example, while the computer-readable medium may be described as a single medium, the term “computer-readable medium” includes a single medium or multiple media, such as a centralized or distributed database, and/or associated caches and servers that store one or more sets of instructions. The term “computer-readable medium” shall also include any medium that is capable of storing, encoding or carrying a set of instructions for execution by a processor or that cause a computer system to perform any one or more of the embodiments disclosed herein.

[0079] The computer-readable medium may comprise a non-transitory computer-readable medium or media and/or comprise a transitory computer-readable medium or media. In a particular non-limiting, exemplary embodiment, the computer-readable medium can include a

solid-state memory such as a memory card or other package that houses one or more non-volatile read-only memories. Further, the computer-readable medium can be a random access memory or other volatile re-writable memory. Additionally, the computer-readable medium can include a magneto-optical or optical medium, such as a disk or tapes or other storage device to capture carrier wave signals such as a signal communicated over a transmission medium. Accordingly, the disclosure is considered to include any computer-readable medium or other equivalents and successor media, in which data or instructions may be stored.

[0080] Although the present application describes specific embodiments which may be implemented as computer programs or code segments in computer-readable media, it is to be understood that dedicated hardware implementations, such as application specific integrated circuits, programmable logic arrays and other hardware devices, can be constructed to implement one or more of the embodiments described herein. Applications that may include the various embodiments set forth herein may broadly include a variety of electronic and computer systems. Accordingly, the present application may encompass software, firmware, and hardware implementations, or combinations thereof. Nothing in the present application should be interpreted as being implemented or implementable solely with software and not hardware.

[0081] Although the present specification describes components and functions that may be implemented in particular embodiments with reference to particular standards and protocols, the disclosure is not limited to such standards and protocols. Such standards are periodically superseded by faster or more efficient equivalents having essentially the same functions. Accordingly, replacement standards and protocols having the same or similar functions are considered equivalents thereof.

[0082] The illustrations of the embodiments described herein are intended to provide a general understanding of the various embodiments. The illustrations are not intended to serve as a complete description of all of the elements and features of apparatus and systems that utilize the structures or methods described herein. Many other embodiments may be apparent to those of skill in the art upon reviewing the disclosure. Other embodiments may be utilized and derived from the disclosure, such that structural and logical substitutions and changes may be made without departing from the scope of the disclosure. Additionally, the illustrations are merely representational and may not be drawn to scale. Certain proportions within the illustrations may be exaggerated, while other proportions may be minimized. Accordingly, the disclosure and the figures are to be regarded as illustrative rather than restrictive.

[0083] One or more embodiments of the disclosure may be referred to herein, individually and/or collectively, by the term “invention” merely for convenience and without intending to voluntarily limit the scope of this application to any particular invention or inventive concept. Moreover, although specific embodiments have been illustrated and described herein, it should be appreciated that any subsequent arrangement designed to achieve the same or similar purpose may be substituted for the specific embodiments shown. This disclosure is intended to cover any and all subsequent adaptations or variations of various embodiments. Combinations of the above embodiments, and other embodiments not specifically described herein, will be apparent to those of skill in the art upon reviewing the description.

[0084] The Abstract of the Disclosure is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In addition, in the foregoing Detailed Description, various features may be grouped together or described in a single embodiment for the purpose of streamlining the disclosure. This disclosure is not to be interpreted as reflecting an intention that the claimed embodiments require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter may be directed to less than all of the features of any of the disclosed embodiments. Thus, the following claims are incorporated into the Detailed Description, with each claim standing on its own as defining separately claimed subject matter.

[0085] The above disclosed subject matter is to be considered illustrative, and not restrictive, and the appended claims are intended to cover all such modifications, enhancements, and other embodiments which fall within the true spirit and scope of the present disclosure. Thus, to the maximum extent allowed by law, the scope of the present disclosure is to be determined by the broadest permissible interpretation of the following claims and their equivalents, and shall not be restricted or limited by the foregoing detailed description.

What is claimed is:

1. A method for detecting malware, the method being implemented by a processor on a computing device, the method comprising:
 - receiving, by the processor, a request for a first web page;
 - determining, by the processor, at least one expected attribute of the first web page;
 - generating, by the processor, a modified web page by combining the first web page with a predetermined set of code, the predetermined set of code being configured to detect whether the malware is present on a client device;
 - transmitting, to the client device, the modified web page;
 - receiving, from the client device, data collected from an executed version of the modified web page;
 - determining, by the processor, at least one attribute of the executed version of the modified web page;
 - comparing, by the processor, the at least one expected attribute of the first web page with the at least one attribute of the executed version of the modified web page; and
 - determining, by the processor, whether the malware is present on the client device based on a result of the comparing.
2. The method of claim 1, further comprising transmitting, to the client device, an alert message indicating that the malware is present on the client device.
3. The method of claim 1, wherein the at least one expected attribute of the first web page includes a first number of JavaScript elements that are associated with the first web page, and wherein the determining whether the malware is present on the client device further comprises comparing the first number of JavaScript elements with a second number of JavaScript elements that are associated with the executed version of the modified web page.
4. The method of claim 1, wherein the at least one expected attribute of the first web page includes a first set of identifiers that correspond to JavaScript elements that are associated with the first web page, and wherein the determining whether the malware is present on the client device further comprises comparing the first set of identifiers with a second set of identifiers that correspond to JavaScript elements that are associated with the executed version of the modified web page.

5. The method of claim 1, wherein the at least one expected attribute of the first web page includes a first set of sources of JavaScript elements that are associated with the first web page, and wherein the determining whether the malware is present on the client device further comprises comparing the first set of sources with a second set of sources of JavaScript elements that are associated with the executed version of the modified web page.

6. The method of claim 1, wherein the at least one expected attribute of the first web page includes a first set of Uniform Resource Identifiers (URIs) that are associated with the first web page, and wherein the determining whether the malware is present on the client device further comprises comparing the first set of URIs with a second set of URIs that are associated with the executed version of the modified web page.

7. The method of claim 1, wherein the at least one expected attribute of the first web page includes a first set of JavaScript function names that are associated with the first web page, and wherein the determining whether the malware is present on the client device further comprises comparing the first set of JavaScript function names with a second set of JavaScript function names that are associated with the executed version of the modified web page.

8. The method of claim 1, wherein the at least one expected attribute of the first web page includes first information that relates to JavaScript event handlers that are associated with the first web page, and wherein the determining whether the malware is present on the client device further comprises comparing the first information with second information that relates to JavaScript event handlers that are associated with the executed version of the modified web page.

9. The method of claim 1, wherein the at least one expected attribute of the first web page includes a first set of cookies that are associated with the first web page, and wherein the determining whether the malware is present on the client device further comprises comparing the first set of cookies with a second set of cookies that are associated with the executed version of the modified web page.

10. The method of claim 1, further comprising:
when a determination is made that malware is not present on the client device, storing information that relates to the first web page and the client device in a first database; and

when a determination is made that malware is present on the client device, storing information that relates to the first web page, the client device, and the malware in a second database.

11. A computing device configured to implement an execution of a method for detecting malware, the computing device comprising:

- a display screen;
- a processor;
- a memory; and
- a communication interface coupled to each of the processor, the memory, and the display screen,

wherein, when the method is being executed, the processor is configured to:

- receive, via the communication interface, a request for a first web page;
- determine at least one expected attribute of the first web page;
- generate a modified web page by combining the first web page with a predetermined set of code, the predetermined set of code being configured to detect whether the malware is present on a client device;
- transmit, to the client device via the communication interface, the modified web page;
- receive, from the client device via the communication interface, data collected from an executed version of the modified web page;
- determine at least one attribute of the executed version of the modified web page;
- compare the at least one expected attribute of the first web page with the at least one attribute of the executed version of the modified web page; and
- determine whether the malware is present on the client device based on a result of the comparison.

12. The computing device of claim 11, wherein the processor is further configured to transmit, to the client device via the communication interface, an alert message indicating that the malware is present on the client device.

13. The computing device of claim 11, wherein the at least one expected attribute of the first web page includes a first number of JavaScript elements that are associated with the first web page, and wherein the processor is further configured to determine whether the malware is present on the client device by comparing the first number of JavaScript elements with a second

number of JavaScript elements that are associated with the executed version of the modified web page.

14. The computing device of claim 11, wherein the at least one expected attribute of the first web page includes a first set of identifiers that correspond to JavaScript elements that are associated with the first web page, and wherein the processor is further configured to determine whether the malware is present on the client device by comparing the first set of identifiers with a second set of identifiers that correspond to JavaScript elements that are associated with the executed version of the modified web page.

15. The computing device of claim 11, wherein the at least one expected attribute of the first web page includes a first set of sources of JavaScript elements that are associated with the first web page, and wherein the processor is further configured to determine whether the malware is present on the client device by comparing the first set of sources with a second set of sources of JavaScript elements that are associated with the executed version of the modified web page.

16. The computing device of claim 11, wherein the at least one expected attribute of the first web page includes a first set of Uniform Resource Identifiers (URIs) that are associated with the first web page, and wherein the processor is further configured to determine whether the malware is present on the client device by comparing the first set of URIs with a second set of URIs that are associated with the executed version of the modified web page.

17. The computing device of claim 11, wherein the at least one expected attribute of the first web page includes a first set of JavaScript function names that are associated with the first web page, and wherein the processor is further configured to determine whether the malware is present on the client device by comparing the first set of JavaScript function names with a second set of JavaScript function names that are associated with the executed version of the modified web page.

18. The computing device of claim 11, wherein the at least one expected attribute of the first web page includes first information that relates to JavaScript event handlers that are associated with the first web page, and wherein the processor is further configured to determine whether the malware is present on the client device by comparing the first information with

second information that relates to JavaScript event handlers that are associated with the executed version of the modified web page.

19. The computing device of claim 11, wherein the at least one expected attribute of the first web page includes a first set of cookies that are associated with the first web page, and wherein the processor is further configured to determine whether the malware is present on the client device by comparing the first set of cookies with a second set of cookies that are associated with the executed version of the modified web page.

20. The computing device of claim 1, wherein:
when a determination is made that malware is not present on the client device, the processor is further configured to store, in the memory, information that relates to the first web page and the client device in a first database; and
when a determination is made that malware is present on the client device, the processor is further configured to store, in the memory, information that relates to the first web page, the client device, and the malware in a second database.

21. A method for detecting malware, the method being implemented by a processor on a computing device, the method comprising:
transmitting, to a server device, a request for a first web page;
receiving, from the server device and in response to the request, a modified version of the first web page;
executing, by the processor, the received modified version of the first web page; and
transmitting, to the server device, data collected from an executed version of the received modified version of the first web page,
wherein the server device is configured to generate the modified version of the first web page and to use the data collected from the executed version of the received modified version of the first web page to determine whether the malware is present on the computing device.

22. The method of claim 21, further comprising receiving, from the server device when the server device determines that malware is present on the computing device, an alert message indicating that the malware is present on the computing device.