



- (51) International Patent Classification:
H04L 29/06 (2006.01)
- (21) International Application Number:
PCT/US2017/017293
- (22) International Filing Date:
10 February 2017 (10.02.2017)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
62/295,315 15 February 2016 (15.02.2016) US
62/314,255 28 March 2016 (28.03.2016) US
62/328,912 28 April 2016 (28.04.2016) US
- (72) Inventor; and
(71) Applicant : **WOOD, Michael** [US/US]; 6 N.E. 24th
Street, Lazy Lake, Florida 33305 (US).
- (74) Agent: **JOHNSON, James David**; Johnson & Martin,
P.A., 500 West Cypress Creek Road, Suite 430, Fort Laud-
erdale, Florida 33309 (US).
- (81) Designated States (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN,
KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA,
MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG,
NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS,
RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY,
TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN,
ZA, ZM, ZW.

- (84) Designated States (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,
TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,
TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,
DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *without international search report and to be republished
upon receipt of that report (Rule 48.2(g))*

(54) Title: SYSTEM AND METHOD FOR BLOCKING PERSISTENT MALWARE

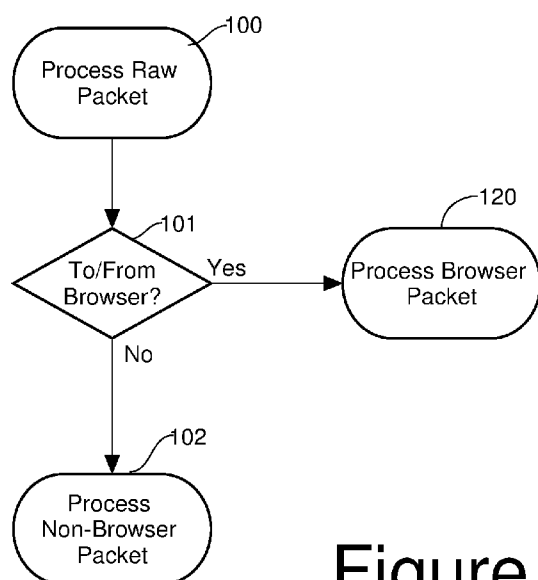


Figure 1

(57) Abstract: Systems and methods for allowing and blocking data packets sent to and from browser software applications and non-browser software applications operated on a computing device are described. The systems can use whitelisting based on the maker of a non-browser software application being a trusted source of data packets sent to the non-browser application or based on consensus by one or more consensus trusted computing devices that connections made by an URL open in a browser software application are correct and trusted. The system uses a firewall to block data packets to and from web addresses that are not owned by the maker of the application, are not trusted by the consensus trusted computing devices, or are blocked by selection by a user of the computing device. The system can also include a health monitor engine to ensure a kernel drive of the system is operational and not disabled by malware.

A great need exists to finally be able to expose and block persistent malware. A further need exists to be able to detect and block Internet traffic to and from Trojans with which a computing device is infected while ensuring that the legitimate traffic to and from the infected apps is allowed.

5

SUMMARY

The invention relates to systems and methods for detecting and blocking data packets sent to and from browser software applications and/or non-browser software applications installed on a computing device and infected with Trojans and/or other malware that attempts to communicate with one or more hacker's command and control centers. In exemplary embodiments, the system is capable of allowing and blocking data packets sent to and from both browser software applications and non-browser software applications installed on a computing device. In other embodiments, systems that allow and block traffic to and from only browser software applications or to and from only non-browser software applications may be used, although for providing the most effective computer security systems that perform both methods are used. The allowance and blocking of data packets to and from various sources may be performed automatically by the system or may be accomplished by action taken by a user of the computing device.

The systems and methods described herein can use whitelisting based on the maker of a non-browser software application being a trusted source of data packets sent to the non-browser application or based on consensus by one or more consensus trusted computing devices that connections made by an URL open in a browser software application are correct and trusted. The system uses a local firewall of the computing device to block data packets to and from web addresses (i.e., domains, subdomains, path names, or URLs) that are not owned by the makers of the application, that are not trusted by the consensus trusted computing devices, or that are blocked by affirmative selection for blocking by a user of the computing device. The system can also include a health monitor engine to ensure that a kernel drive of the system is operational and not disabled by malware, which would prevent the system from detecting and blocking traffic between the browser and non-browser

software applications installed on a computing device and a hacker's command and control center.

Accordingly, the invention features a system for detecting and blocking Trojan malware and other malware as follows: The system includes a computing
5 device that is communicatively connected to a communications network, and a browser software application being operated on the computing device. The system also includes an open URL identification process to communicate an identity of a web address, wherein the web address is an open web address if it is open in the address bar of the browser software application. The system further includes an
10 open connection tracking process that tracks at least one connection to a remote computing device made by the open web address. The open connection tracking process can report open connections as one or more of the following: domains, subdomains, path names, resource locations, URLs, IP addresses, and the like. The system also includes a firewall to intercept data packets sent to or from the browser
15 software application. The firewall blocks data packets that are not being sent to or from the open web address or to or from the at least one connection of the open web address.

In another aspect, the invention can feature the web address being a URL name, a path name of a URL, a domain name of a URL, or a subdomain of a URL.
20 The identity can be an owner name, a URL name, a path name of a URL, a domain name of a URL, a subdomain of a URL, a derivative of one of the foregoing, or an alias representing one of the foregoing.

In another aspect, the invention can feature the at least one connection, if any, made by the web address being a URL name, a path name of a URL, a resource
25 named by a URL, a domain name of a URL, or a subdomain of a URL. The identity can be an owner name, a URL name, a path name of a URL, a resource named by a URL, a domain name of a URL, a subdomain of a URL, a derivative of one of the foregoing, or an alias representing one of the foregoing

In another aspect, the invention can feature the connection tracking process
30 verifying the authenticity of the at least one connection made by the open web address. The system further includes at least one trusted computing device to

identify at least authentic one open connection of the open web address. The at least one trusted computing device reports an authenticity list that includes at least one connection of the web address that is authentic.

5 In another aspect, the invention can feature a connection analysis engine to statistically analyze the connections reported by multiple computing devices that access the same open web address to verify that the at least one connection is a trusted connection that the connection analysis engine assigns a trusted status or is not a trusted connection that the connection analysis engine assigns a distrusted status.

10 In another aspect, the invention can feature the connection analysis engine being operated on the computing device or on a different computing device that is communicatively connected to the computing device.

In another aspect, the invention can feature the firewall being configured to intercept and automatically block data packets sent to or from the browser software
15 application that are not being sent to or from the open web address or to or from the at least one connection of the open web address.

In another aspect, the invention can feature a reputation engine to allow or block the open web address. The firewall blocks the open web address if the open web address does not meet a reputation threshold as determined by the reputation
20 engine.

In another aspect, the invention can feature a reputation engine to allow or block the at least one connection made by the open web address. The firewall blocks the at least one connection made by the open web address if the at least one connection made by the open web address does not meet a reputation threshold as
25 determined by the reputation engine.

In another aspect, the invention can feature a reputation engine to allow or block at least one other connection to the browser software application. The firewall blocks the at least one other connection if the at least one other connection does not meet a reputation threshold as determined by the reputation engine. The at least one
30 other connection includes one or more connections that are not the open web address or the at least one connection to the open web address.

In another aspect, the invention can feature a web address identity interface to display the identity of the open web address to a user, wherein the web address identity interface comprises at least one control element that permits the user to allow or block data packets sent to and from the open web address by selecting an
5 allowed status or a blocked status for the open web address. The at least one connection, if any, of the open web address is also assigned the blocked status when the user operates the at least one control element of the web address identity interface to block the open web address. The firewall blocks data packets sent to and from the at least one connection, if any.

10 In another aspect, the invention can feature the at least one connection of the open web address being allowed, and not blocked, when the open web address is blocked but a second open web address that is allowed shares the same at least one connection.

In another aspect, the invention can feature a connection interface that
15 displays the at least one connection of the open web address. The connection interface includes at least one control element that permits the user to allow or block the at least one connection. The firewall blocks data packets sent to and from the at least one connection if the blocked status is selected for the at least one connection. The connection interface can be the web address identity interface or a different
20 interface, and the at least one control element of the connection interface can be the at least one control element of the web address identity interface or a different at least one control element.

In another aspect, the invention can feature the interface displaying the identity of the web address of automatically blocked data packets. The blocked
25 status of the web address of the automatically blocked data packets is overridable by at least one option selected from among the group of: (i) the user using the at least one control element of the interface to manually select the allowed status for the web address to override the blocked status of the web address; (ii) an override process overriding the blocked status of the web address by comparison to a whitelist that
30 informs the override process that the data packets are not being sent to or from the open web address or its at least one connection; or (iii) a reputation engine overriding

the blocked status of the web address when the data packets are not being sent to or from the open web address or its at least one connection when a reputation threshold is met or exceeded as determined by the reputation engine.

In another aspect, the invention can feature the allowed open web address
5 being a user-opened web address opened by the user. The system further includes an interface to display the identity of the user-opened web address to a user. The at least one connection includes at least one connection to a remote computing device made by the user-opened web address, if any. The interface does not display the at least one connection made by the user-opened web address but does allow data
10 packets sent to or from that at least one connection. The interface displays and the firewall blocks data packets that are not sent to or received from the browser software application by the user-opened web address or its at least one connection.

In another aspect, the invention can feature an override process to override
the blocked status of the data packets that are not sent to or received from the browser
15 software application by the user-opened web address or its at least one connection. The override process compares a web address or identity of the blocked data packets to a whitelist that informs the override process that the blocked status of the blocked data packets should be overridden. The override process displays the blocked data packets and changes their blocked status to allowed status if the web address or
20 identity of the blocked data packets is located in the whitelist.

In another aspect, the invention can feature a web traffic analysis system to determine the identity of the open web address, the identity of the at least one connection of the open web address, or the identity of both. The web traffic analysis system includes a browser plugin, a local man-in-the-middle http/https interception,
25 a remote man-in-the-middle http/https interception, or a web proxy.

In another aspect, the invention can feature an automated health monitor engine that periodically checks a status of a kernel driver of the system to ensure that the kernel driver is operational.

In another aspect, the invention can feature the automated health monitor
30 engine generating an alert if the kernel driver is disabled.

In another aspect, the invention can feature an interface having a continuously changing visual display. A continuous change in appearance of the continuously changing visual display represents that the automated health monitor engine is checking the kernel driver. The continuously changing visual display
5 ceases continuously changing in appearance if the automated health monitor engine is unable to perform the check of the status of the kernel driver.

The invention also features a system for detecting and blocking Trojan malware and other malware as follows: The system includes a computing device that is communicatively connected to a communications network, and a browser
10 software application being operated on the computing device. The system also includes a remote web proxy that replaces at least one connection made by an open web address, if any, with an alias based on a domain name of the web proxy or another domain name. The system further includes a firewall that blocks all data packets not sent to or received from the web proxy by the browser software
15 application. The web address is an open web address if it is open in the web proxy.

In another aspect, the invention can feature the web proxy reporting the open web address to the firewall.

In another aspect, the invention can feature a firewall interface displaying the open web address.

20 In another aspect, the invention can feature the firewall interface permitting a user to select an allowed status or a blocked status for the web address. The firewall communicates the web address to the web proxy when a blocked status is selected for the web address. The web proxy subsequently blocks data packets sent to or received from the web address and the web proxy blocks the connections made
25 by the web address, if any.

In another aspect, the invention can feature data packets not sent to or received from the web proxy by the browser software application are initially automatically blocked except data packets sent to or received from the browser software application by a web address or IP address that appears on a whitelist.

30 The invention also features a system for detecting and blocking Trojan malware and other malware as follows: The system includes a computing device that

is communicatively connected to a communications network, and at least one non-browser software application operated on the computing device. The system also includes a maker identification process to identify a maker of the at least one non-browser software application. The system further includes an identity verification
5 process to determine an owner of at least one remote host address sending data packets to or receiving data packets from each at least one non-browser software application. The owner of each at least one remote host address as determined by the identity verification process and the maker of each at least one non-browser software application are compared to determine whether they are the same or
10 different. The system also includes a firewall to block data packets sent to and from the at least one non-browser software application by each at least one remote host address when the owner of each at least one remote host address and the maker of each at least one non-browser software application are different.

In another aspect, the invention can feature a mismatched application/remote
15 host address communication pair being one at least one remote host address having an owner that is different from the maker of one at least one non-browser software application. The identification of a mismatched application/remote host address communication pair results in the firewall blocking data packets transmitted between the mismatched application/remote host address pair.

20 In another aspect, the invention can feature a reputation engine that initially blocks data packets from the at least one remote host address when a reputation of the at least one remote host address is determined by the reputation engine to not meet or exceed a reputation threshold.

In another aspect, the invention can feature the system having a first interface
25 to display a name of each at least one non-browser software application, the maker of each at least one non-browser software application, each at least one remote host address with which each at least one non-browser application attempts to communicate, and the owner of each at least one remote host address. The system can also include a second interface that permits a user to selectively allow or block
30 traffic sent to and received from each at least one non-browser software application

by each at least one remote host address. The second interface can be the first interface or a different interface.

In another aspect, the invention can feature each non-browser software application of the at least one non-browser software application that communicating
5 with a remote host address of the at least one web address is an application/remote host address communication pair. The system includes a process to initially block all application/remote host address communication pairs. The system further includes a first interface to display a name of each non-browser software application, the maker of each non-browser software application, each web address with which
10 each non-browser software application communicates, and the owner of each such web address. The system can also include a second interface that permits a user to selectively allow or block traffic sent to and received from each at least one non-browser software application by each at least one web address. The second interface is the first interface or a different interface.

15 In another aspect, the invention can feature each non-browser software application of the at least one non-browser software application that communicating with a remote host address of the at least one remote host address is an application/remote host address communication pair. The system includes a process to initially block traffic between all application/remote host address communication
20 pairs for a temporary period of time. The system further includes a first interface to display a name of each non-browser software application, the maker of each non-browser software application, each remote host address with which each non-browser software application communicates, and the owner of each such remote host address. The system further includes a second interface that permits a user to
25 selectively quarantine one or more application/remote host address communication pairs of all of the application/remote host address communication pairs prior to expiration of the temporary period of time. The second interface can be the first interface or a different interface.

In another aspect, the invention can feature legitimate DNS data packets and local
30 packets and traffic to digital certificate authorities being automatically allowed.

The invention also features a system for detecting and blocking Trojan malware and other malware as follows: The system includes a computing device that is communicatively connected to a communications network, and at least one non-browser software application operated on the computing device. The system also
5 includes an identity verification process to determine an owner of at least one remote host address sending data packets to or receiving data packets from each at least one non-browser software application. The owner of each at least one remote host address as determined by the identity verification process and the maker of each at least one non-browser software application can be compared to determine whether
10 they are the same or different. The system further includes an interface to communicate both an identity of the at least one non-browser software application and the owner of the at least one remote host address with which the at least one non-browser software application is communicating. The system further includes a firewall to block data packets transmitted between at least one
15 application/destination pair, wherein the at least one application/destination pair includes one of the at least one non-browser applications communicating with one of the at least one remote host addresses that is the destination of the at least one application/destination pair.

In another aspect, the invention can feature an interface to permit a user to
20 selectively allow or block one or more of the at least one application/destination pairs.

In another aspect, the invention can feature the firewall automatically blocking communication between each newly encountered application/destination pair of the at least one application/destination pair. The interface permits the user to
25 toggle a current status of each at least one application/destination pair between an allowed status and a blocked status.

In another aspect, the invention can feature the firewall automatically blocking communication between each newly encountered application/destination pair of the at least one application/destination pair unless the newly encountered
30 application/destination pair, the destination, or both are whitelisted. The interface

permits the user to toggle a current status of each at least one application/destination pair between an allowed status and a blocked status.

In another aspect, the invention can feature the firewall automatically blocking communication between each newly encountered application/destination pair of the at least one application/destination pair unless a destination reputation of the destination meets or exceeds a destination reputation threshold. The system further includes a destination reputation engine to determine the destination reputation of the destination. The interface permits the user to toggle a current status of each at least one application/destination pair between an allowed status and a blocked status.

Unless otherwise defined, all technical terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this invention belongs. Although methods and materials similar or equivalent to those described herein can be used in the practice or testing of the present invention, suitable methods and materials are described below. All publications, patent applications, patents and other references mentioned herein are incorporated by reference in their entirety. In the case of conflict, the present specification, including definitions will control.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a flow chart showing a process of the system for processing data packets.

Figure 2 is a flow chart showing one embodiment of a system for processing data packets sent and received by a browser app.

Figure 3 is a flow chart showing one embodiment of a system for processing data packets sent and received by a non-browser app.

Figure 4 is a flow chart showing another embodiment of a system for processing data packets data packets sent and received by a browser app.

Figure 5 is a flow chart showing still another embodiment of a system for processing data packets data packets sent and received by a browser app.

Figure 6 is a flow chart showing another embodiment of a system for processing data packets data packets sent and received by a non-browser app.

Figure 7 is a flow chart showing still another embodiment of a system for processing data packets data packets sent and received by a non-browser app.

5 Figure 8 is a graphic representation of the principles that allow the system to operate effectively.

Figure 9 is a schematic diagram of one embodiment of the system.

DETAILED DESCRIPTION

The present invention is best understood by reference to the detailed
10 drawings and description set forth herein. Embodiments of the invention are discussed below with reference to the drawings; however, those skilled in the art will readily appreciate that the detailed description given herein with respect to these figures is for explanatory purposes as the invention extends beyond these limited embodiments. For example, in light of the teachings of the present invention, those
15 skilled in the art will recognize a multiplicity of alternate and suitable approaches, depending upon the needs of the particular application, to implement the functionality of any given detail described herein beyond the particular implementation choices in the following embodiments described and shown. That is, numerous modifications and variations of the invention may exist that are too
20 numerous to be listed but that all fit within the scope of the invention. Also, singular words should be read as plural and vice versa and masculine as feminine and vice versa, where appropriate, and alternative embodiments do not necessarily imply that the two are mutually exclusive.

The present invention should not be limited to the particular methodology,
25 compounds, materials, manufacturing techniques, uses, and applications, described herein, as these may vary. The terminology used herein is used for the purpose of describing particular embodiments only, and is not intended to limit the scope of the present invention. As used herein and in the appended claims, the singular forms “a,” “an,” and “the” include the plural reference unless the context clearly dictates
30 otherwise. Thus, for example, a reference to “an element” is a reference to one or more elements and includes equivalents thereof known to those skilled in the art.

Similarly, for another example, a reference to “a step” or “a means” may be a reference to one or more steps or means and may include sub-steps and subservient means.

5 All conjunctions used herein are to be understood in the most inclusive sense possible. Thus, a group of items linked with the conjunction “and” should not be read as requiring that each and every one of those items be present in the grouping, but rather should be read as “and/or” unless expressly stated otherwise. Similarly, a group of items linked with the conjunction “or” should not be read as requiring mutual exclusivity among that group, but rather should be read as “and/or” unless
10 expressly stated otherwise. Structures described herein are to be understood also to refer to functional equivalents of such structures. Language that may be construed to express approximation should be so understood unless the context clearly dictates otherwise.

Unless otherwise defined, all terms (including technical and scientific terms)
15 are to be given their ordinary and customary meaning to a person of ordinary skill in the art, and are not to be limited to a special or customized meaning unless expressly so defined herein.

Terms and phrases used in this application, and variations thereof, especially in the appended claims, unless otherwise expressly stated, should be construed as
20 open ended as opposed to limiting. As examples of the foregoing, the term “including” should be read to mean “including, without limitation,” “including but not limited to,” or the like; the term “having” should be interpreted as “having at least”; the term “includes” should be interpreted as “includes but is not limited to”; the term “example” is used to provide exemplary instances of the item in discussion,
25 not an exhaustive or limiting list thereof; and use of terms like “preferably,” “preferred,” “desired,” “desirable,” or “exemplary” and words of similar meaning should not be understood as implying that certain features are critical, essential, or even important to the structure or function of the invention, but instead as merely intended to highlight alternative or additional features that may or may not be
30 utilized in a particular embodiment of the invention.

Those skilled in the art will also understand that if a specific number of an introduced claim recitation is intended, such an intent will be explicitly recited in the claim, and in the absence of such recitation no such intent is present. For example, as an aid to understanding, the appended claims may contain usage of the

5 introductory phrases “at least one” and “one or more” to introduce claim recitations; however, the use of such phrases should not be construed to imply that the introduction of a claim recitation by the indefinite articles “a” or “an” limits any particular claim containing such introduced claim recitation to embodiments containing only one such recitation, even when the same claim includes the

10 introductory phrases “one or more” or “at least one” and indefinite articles such as “a” or “an” (e.g., “a” and “an” should typically be interpreted to mean “at least one” or “one or more”); the same holds true for the use of definite articles used to introduce claim recitations. In addition, even if a specific number of an introduced claim recitation is explicitly recited, those skilled in the art will recognize that such

15 recitation should typically be interpreted to mean at least the recited number (e.g., the bare recitation of “two recitations,” without other modifiers, typically means at least two recitations, or two or more recitations). Furthermore, in those instances where a convention analogous to “at least one of A, B, and C” is used, in general, such a construction is intended in the sense one having skill in the art would

20 understand the convention (e.g., “a system having at least one of A, B, and C” would include but not be limited to systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together, etc.).

All numbers expressing dimensions, quantities of ingredients, reaction conditions, and so forth used in the specification are to be understood as being

25 modified in all instances by the term “about” unless expressly stated otherwise. Accordingly, unless indicated to the contrary, the numerical parameters set forth herein are approximations that may vary depending upon the desired properties sought to be obtained.

The invention provides a novel system and method that both exposes and

30 blocks persistent malware – finally providing genuinely effective protection against this now most prevalent form of hacking. Trojans are but one example of persistent

malware that communicates with a hacker's command and control center; other types of persistent malware exist as will be understood by one of skill in the field. As used herein, the terms "computer" and "computing device" mean a computer, a tablet computer, a server, a mobile device (e.g., a cellular telephone, smartphone, etc.), a gaming device, a smart appliance, a smart television, a smart light bulb, a smart outlet adapter, a smart automobile, and any other device that includes a processor and is communicatively connected to a communications network. The term "smart" used with several items in the foregoing list means that such items are electronic devices that have a wireless connection to a communications network.

For purposes of convenience, the term "computing device" is used most frequently herein. In most but not all instances, the computing device also includes at least one browser software application, at least one non-browser software application, or both. Such software applications can be installed locally on the computing device or can be installed on a remote computing device but operated on the local computing device.

The term "web address" used herein means a URL name, a path name of a URL, a domain name of a URL, or a subdomain of a URL. The terms "identity" and "name," when used alone herein, mean an owner name, a URL name, a path name of a URL, a domain name of a URL, a subdomain of a URL, a derivative of one of the foregoing, or an alias representing one of the foregoing.

The invention features a system for detecting and blocking Trojans and other persistent malware that connect to one or more hackers' command and control centers. The system includes a browser software application installed on a computing device that is communicatively connected to a communications network, a process to communicate an identity or identities of one or more web addresses open in the browser software application, a process that catalogues the connections made by open web addresses, a firewall that intercepts packets to or from a browser and blocks packets that do not belong to an open web address or to any of the connections made by an open web address.

The system can feature automatically blocking a packet if it is to or from a browser yet does not belong to a subdomain of an open web address nor any of the

connections made by an open web address. One or more trusted computing devices can record the correct connections made by web addresses so that the connections can be verified before being trusted.

The system can include a process that uses a statistical analysis to deduce the correct connections made by web addresses so that the connections can be verified before being trusted. The system can also include an interface, which displays the names of open web addresses to the user in a manner that permits the user to allow or block the web address. Such names can be the owner of the domain, a domain name, subdomain name, and/or a derivative of these and/or an alias representing any of these. When the user blocks a web address then all of its connections are blocked (unless such a connection is made by a separate, currently allowed web address).

The interface can display the connections opened by web addresses. The interface can display the connections opened by web addresses in a manner that permits the user to allow or deny such connections. The user may elect to allow only data packets that belong to user-opened web addresses or connections made by user-opened web addresses that the user has not denied.

The system can also include a reputation engine that can be used to initially allow or deny a web address (and its subsequent connections), such that web addresses that do not meet an established reputation threshold are initially blocked rather than initially allowed. In another embodiment, the reputation engine can be used to initially allow or deny a web address (and its subsequent connections), such that domains and/or subdomains that do not meet an established reputation threshold are initially blocked rather than initially allowed.

The aforementioned interface (or a different interface of the system) can display the names of automatically blocked packets. Such names can be a name of the owner of the domain, a domain name, subdomain name, and/or a derivative of these and/or an alias representing any of these. In another embodiment, the interface can display the names of automatically blocked packets in a manner that permits the user to override the blocked name. As with the prior embodiment, such names can be the owner of the domain, a domain name, subdomain name, and/or a derivative of these and/or an alias representing any of these.

In another embodiment, the system can feature a whitelist that can override the automatic blocking of browser-based packets that neither belong to an open web address nor to any of the connections made by open web addresses.

5 The reputation engine of the system may also be capable of overriding the automatic blocking of browser-based packets that neither belong to an open web address nor to any of the connections made by open web addresses if the reputation meets or exceeds an established threshold.

10 The system may display and allow the names of user-opened web addresses while not displaying, yet allowing, data packets belonging to connections made by user-opened web addresses. In this embodiment, the system may display and block the names of packets that do not meet the prior two conditions.

15 In another embodiment, the system may display and allow the names of user-opened web addresses while not displaying, yet allowing, data packets belonging to connections made by user-opened web addresses. The names of web addresses sending and receiving data packets that do not meet the prior two conditions may be displayed and blocked unless such packets are whitelisted in which case they are displayed and allowed.

20 The system can also feature an automated health monitor engine that periodically checks the status of the kernel driver to ensure that the kernel driver is operational. The automated health monitor engine can generate an alert if the kernel driver is disabled. The alert can be visual, audible, or both. In an exemplary embodiment, the automated health monitor engine can include an interface with a constantly changing visual display representing the health monitor checking of the kernel driver such that the display will cease constantly changing if the user interface is prevented from performing the kernel driver health monitoring, thereby alerting the user that the system and computing device have potentially been hacked.

25 The system can include a browser plugin, man-in-the-middle http/https interception (local and/or remote), and/or a web proxy for determining the identity of open web addresses and of connections made by open web addresses.

30 The invention also features a system for detecting and blocking Trojans and other malware that communicate with one or more hackers' command and control

centers. The system includes a browser software application installed on a computing device that is communicatively connected to a communications network and a remote web proxy, which replaces the connections made by web addresses with aliases based on the domain of the web proxy or some other agreed upon domain. The system also includes a local firewall that blocks all browser-based packets not destined to or received from the web proxy. The web proxy can report the open web address to the firewall, and a firewall interface can display the open web addresses.

The firewall interface can permit the user to allow or deny the web addresses. The local firewall communicates denied web addresses to the web proxy, and the web proxy subsequently blocks the affected web addresses and their connections.

The system may initially block data packets that are not destined to or received from the web proxy except whitelisted subdomains, domains, domain owners, and/or IP addresses (which are allowed).

In one example, a web proxy could be used for browser traffic in which the web proxy communicates the web addresses of one or more open web addresses to the firewall of the system, and the web proxy could replace the connections made by the open web addresses with aliases such that the firewall could automatically block data packets if they do not belong to the subdomain and/or domain of any open web address.

The invention also features a system for detecting and blocking Trojans and other malware that connect to one or more hackers' command and control centers, wherein the system includes a non-browser software application installed on a computing device that is communicatively connected to a communications network, a process that communicates the owner of the domains of the data packets sent to or from the non-browser app, a process to communicate the maker of the non-browser app, and a process that initially blocks packets when the identity of the maker of the app does not match the identity of the owner of the data packet's domain.

This system can include a reputation engine in which packets are only initially blocked if the maker of the app does not match the owner of the domain, or

if the reputation of the domain and/or the domain owner does not meet or exceed an established reputation threshold.

This system can also feature an interface that displays the app names and/or the names of apps' makers, and the owner of the remote domains with which each
5 app attempts to communicate. The system can also include an interface (which may the same interface as the previously described one or a different interface) that permits the user to selectively allow or deny traffic to and from each app on a per domain and/or per domain owner basis.

The system can include a process that initially blocks all app/domain
10 communication pairs, i.e., non-browser software applications and remote host addresses (also referred to herein as "destinations") that communicate or attempt to communicate with one another. An interface of the system can display the app names and/or the names of their makers, and the owner of the remote domains with which they attempt to communicate. The system may include a second interface
15 that permits the user to selectively allow traffic to and from each app on a per domain and/or per domain owner basis, or these steps may be completed using the first interface.

The system can further include a process that initially blocks all app/domain communication pairs for a temporary period of time and an interface that displays
20 the app names and/or the names of their makers, and the owner of the remote domains with which they attempt to communicate. Either that first interface or a second interface may be provided by the system to permit the user to selectively quarantine the app/domain pairs before the time expires for the temporary period of time (in which case, the app/domain pair is allowed if not quarantined before the
25 time expires).

The system may automatically allow legitimate DNS packets, traffic to digital certificate authorities, and legitimate local packets.

As previously detailed in prior disclosures, a user can control who can access the user's computing device when the user views the names of every party who
30 wants to talk, in near real-time, each time the named entity wants to talk (i.e.,

communicate), whether that named entity is currently allowed or not. Entities can be any of the following (or derivatives or aliases of the following):

IP addresses

Subdomains (e.g., a.2mdn.net, b.2mdn.net, y.gstatic.com, and z.gstatic.com)

5 Domains, also referred to herein as Domain Names (e.g., 2mdn.net, and gstatic.net)

Registered Owners (e.g., “Google Inc.”)

We shall refer to the following as “Entity Hierarchy”: IP Addresses → Subdomains → Domains → Registered Owners. Each level of the Entity Hierarchy
10 moving from left to right encompasses more data packets. A single Registered Owner name can encompass hundreds of Domains, thousands of Subdomains, and even tens of thousands IP Addresses. Thus, when displaying the identity of the entity who wants to communicate with the computing device, some embodiments may choose to use Domain Names and/or Registered Owner names to reduce the amount
15 of information a user will need to see in order to make an informed choice over who can access the computing device and who cannot.

Two broad categories of PC internet traffic exists: browser data packets sent to or from a browser software application and non-browser data packets sent to or from non-browser software applications. For non-browser traffic, the user may find
20 it useful to choose the app/destination pair. For example, a user may want to allow WinWord to talk to Registered Owner Microsoft Corp.; yet the same user might want to block WinWord from talking to Registered Owner Richard Smith.

Allowing the user to allow or block an app/destination pair provides genuinely effective protection against the real-world problem of Trojans (also
25 known as Trojan horse malware) injecting themselves into legitimate apps. For example, Richard Smith is the owner of xyz.org. If a Trojan injects itself into WinWord and tries to send sensitive data to trojan.xyz.org, one embodiment could show the user that WinWord wants to talk to Richard Smith. By keeping this connection blocked, yet allowing WinWord to talk to Microsoft Corp., the user’s
30 sensitive information remains protected while WinWord’s legitimate activity remains intact.

Thus, a highly effective method for protecting non-browser apps is to display the name of the app (or a derivative of the name or an alias representing the name) along with the destination name at the time the app wants to communicate with the non-browser app. Moreover, the most secure method would be to initially block
5 each new app/destination pair unless the pair was specifically whitelisted and/or the user chooses to unblock the communication between the pair. In addition to or alternative to displaying the app name would be to display the name of the maker of the app. For example, some embodiments might display that “Microsoft Corp.’s Winword” wants to talk to “Microsoft Corp.”

10 Another highly secure method would be to temporarily block each newly encountered app/destination pair (unless the pair is whitelisted), and to display the requested connection to the user. If the user takes no action then the app/destination pair status automatically changes to allowed. However, if the user takes action then the status changes from “temporarily blocked” to “permanently blocked” (i.e.,
15 blocked until the user specifically decides to allow it and/or the app/destination pair is later added to a whitelist).

A Trojan can literally send millions of bits of data in a single second. Therefore, the initial blocking of the Trojan is important so that not a single bit of data is compromised. On the other hand, some users may not want to have to
20 manually allow every individual app/destination pair. By temporarily blocking all newly encountered app/destination pairs, both objectives can be satisfied for such users, and therefore, some embodiments may implement this particular method.

Showing the name to whom each app wants to talk offers convenient security for non-browser apps since the vast majority of such apps typically talk to a single
25 company, organization, or person. In other words, the vast majority of such non-browser apps typically talk to a single Registered Owner name – usually the name of the maker of the app itself (with notable exceptions being communication with DNS Servers and/or digital certificate authorities). This property of non-browser apps lends itself to a novel form of rule-based whitelisting: if an app wants to talk to
30 its maker and/or a DNS Server and/or a digital certificate authority, then allow it to do so. This rule is hereafter referred to as Maker-Based Whitelisting. This Maker-

Based Whitelisting method elegantly protects the vast majority of non-browser apps when they are hijacked for nefarious hacking purposes.

In stark contrast to non-browser apps, the vast majority of browser traffic is to destinations other than the browser's maker. For example, the vast majority of
5 Firefox traffic is to destinations other than Mozilla (the maker of Firefox). To complicate matters even more, when a user enters in a single URL (e.g. "cnn.com"), that webpage can connect the user to numerous other sites. For example, on January 20, 2017, cnn.com automatically connected users to 56 other sites. Thus, when a Trojan hijacks a browser, the Trojan can easily hide its connection to the command
10 and control center inside of such lists. Most users cannot decide which of the 56 seemingly random connections are legitimate and which one is a Trojan talking to a command and control node.

Therefore, with browsers, even using Registered Owner names (as opposed to domain and subdomains) does not reduce the traffic enough for the average user
15 to spot a Trojan. The previously stated non-browser app whitelisting rule (Maker-Based Whitelisting) does not help either. Browser traffic requires a different approach.

To significantly reduce displayed browser traffic names, traffic can be managed by the following Group Policy: if the user approves a URL (or the URL is
20 whitelisted) then allow all connections made by that URL for as long as its webpage is active in the browser, without displaying such connections. For example, if cnn.com is an allowed URL then automatically allow the 56 sites to which it connects without displaying the 56 sites.

Some embodiments of the system that can accomplish the above include (but
25 are not limited to) a browser plugin that transmits each open URL and their connections to the security system, a local man-in-the-middle http/https interception can be used to retrieve this information, and/or a remote man-in-the-middle http/https proxy can be used to obtain open web addresses and their respective connections.

30 With the Group Policy, the user would see "cnn.com" – not 57 entity names (i.e., "cnn.com" plus the other 56 sites to which it connects), or even more

domain/subdomains. If the user allows cnn.com, then all of cnn.com's connections can be allowed as well.

In some embodiments, a browser plugin can display the connections on the webpages themselves. For example, the main security display of the system would
5 solely show cnn.com while the plugin could show the 56 connections on the cnn.com browser webpage itself. Such embodiments could also give the user the ability to block any and/or all of the URL's connections. Some embodiments might allow the user to see the connections from a Menu choice in the main program and/or control the individual connections from a Menu choice in the browser plugin as well.

10 With the Group Policy, Trojans are now easily exposed and blocked when they hijack the browser. For example, if a Trojan trying to connect to trojan.xyz.org hijacks Firefox while the user is accessing cnn.com, under the Group Policy, the user would see that Firefox wants to connect to two entities: Turner Broadcasting (the registered owner of cnn.com) and Richard Smith (the registered owner of xyz.org).
15 The user can now easily allow the cnn.com connection while keeping the Trojan connection blocked.

The Group Policy can also be used for automation: Automatically allow open web addresses and their connections while automatically blocking everything else. In the example above, cnn.com and its connections would automatically be allowed
20 while Trojan.xyz.org would be automatically blocked (since it is neither an open URL subdomain nor the subdomain of any connection made by an open URL).

The Group Policy easily exposes and blocks Trojans that attempt to make direct connections to their command and control center while masquerading as Firefox. However, an open loophole still remains: the Trojan can inject its command
25 and control center destination into a webpage and thereby cause the connection to be allowed (at least temporarily).

Pages that use pure HTTPS have a degree of protection against such an attack because the HTTPS protocol uses end-to-end encryption to help prevent this very problem. However, many web pages still use HTTP which is unencrypted, and
30 therefore, easily tricked. Also, many HTTPS man-in-the-middle proxies are in use

today, which strip the HTTPS encryption protection. Thus, in neither HTTP nor HTTPS can the contents of the webpages be implicitly trusted.

To close this loophole, some embodiments can use Consensus-Based Whitelisting. In this novel method, the connections made by web addresses are
5 recorded by one or more trusted machines (and/or a statistical deduction of the correct connections can be made from sampling a number of user machines for any given URL). When a user's computing device wants to connect to a URL, the correct connections can be retrieved from the internet via an encrypted connection. These consensus-based connections are then used for the Group Policy (as opposed
10 to inherently trusting the connections listed in the local webpage).

For example, if a Trojan successfully inserts trojan.xyz.org into the local cnn.com webpage (in an effort to get the security system to allow traffic to and from trojan.duckdns.org), under Consensus-Based Whitelisting, the user's computing device will download the 56 correct connections and these alone are used for the
15 Group Policy. Thus, when Firefox seeks to connect to trojan.xyz.org, the user will see both Turner Broadcasting and Richard Smith as the two requested connections (since trojan.xyz.org was not included in the Consensus-Based Whitelisting Group Policy).

Thus, Consensus-Based Whitelisting exposes and blocks Trojans that inject
20 themselves into browsers, regardless of whether they try to make a direct connection or whether they insert their command and control destinations into the local webpages themselves.

An alternative to or adjunct to Consensus-Based Whitelisting is the use of a web proxy for the correct identification of connections made by web addresses. (See
25 below for a description of one embodiment.)

Consensus-Based Whitelisting can also be used for non-browser apps as an adjunct to Maker-Based Whitelisting. As one example, Consensus-Based Whitelisting can be used when an app has legitimate traffic that does not involve its maker. In other words, an app's traffic can be allowed if it passes either the Maker-
30 Based Whitelisting rule or if one or more trusted machines (or a statistically

significant number of other machines) record the same app communicating with the same destination (i.e., Consensus-Based Whitelisting).

Some embodiments can use Consensus-Based Whitelisting to automate the discovery of an apps maker.

5 Remote web proxies can also be used for managing open web addresses and their respective connections. For example, a remote web proxy can report the domain and/or subdomain, and/or path, and/or URL name of each open URL to the firewall. For example, if the user enters “cnn.com” in the web proxy then it can report “cnn.com” as an open URL. The web proxy could also replace connections
10 made by open web addresses with aliases based on the web proxy’s domain or some other agreed upon domain. For example, if the web proxy domain is TerraProxy.com and cnn.com connects to “optimizely.com” and “instantads.com” then the web proxy could replace “optimizely.com” with “abc.TerraProxy.com” and the web proxy could replace “instantads.com” with “xyz.TerraProxy.com”. When
15 the local browser seeks to connect to “abc.TerraProxy.com” the web proxy converts this back to “optimizely.com” and when the local browser seeks to communicate with “xyz.TerraProxy.com” the web proxy converts this back to “instantads.com.” This will force all legitimate web browsing to flow through TerraProxy.com. Therefore, the firewall can initially block all web-based traffic that does not flow
20 through the web proxy (e.g. TerraProxy.com). Moreover, the firewall can provide an interface for the user to see and block the list of open web addresses. The firewall reports the blocked web addresses to the web proxy, which in turn shuts down the web address and its respective connections. This will prevent Trojans from secretly using the web proxy for connectivity to a command and control center.

25 Some embodiments could use a plugin for certain browsers and use the remote web based proxy for other browsers. In other words, the plugin could report open web addresses and manage the connections made by those web addresses for the given browser while the remote web proxy could be used to manage open web addresses and their respective connections for browsers that do not have such plugins
30 available.

The combination of Maker-Based Whitelisting and Consensus-Based Whitelisting (along with Group Policy) provides formidable protection for browser apps and non-browser apps alike. Therefore, a hacker could try to secretly disable the blocking/allowing security in an effort to covertly bypass it. An automated

5 Health Monitor can protect against this remaining attack vector.

Many embodiments will implement the allowing/blocking mechanism as a kernel driver. To determine the health status of the kernel driver, the GUI process can do one or more of the following:

- Periodically check the status of the driver;
- 10 Periodically poll the driver;
- Receive periodic beacons from the driver; and/or
- Consider the driver healthy if traffic data is regularly received from the driver.

For example, in one embodiment the GUI queries the driver every four

15 seconds to get the information it needs to construct the display. Each time the driver responds to the query, this particular embodiment changes the color of a circle on the display (continually transitioning it from a grey color to a blue color and back again). If the driver should fail to respond to the query then the GUI changes the color of the circle to permanent red.

20 In the example embodiment, if a hacker infects the GUI then the color would cease to change, and if the hacker infects the driver then the color changes to red. Such a health monitor alerts the user to both types of infection, thereby protecting the user from all attack vectors.

A number of ways are available to implement such a health monitor.

25 Provided that the health monitor is implemented such that the GUI continually changes the presentation of the health monitor based upon continued observation of driver state within the confines of a real-time traffic monitor integrated with a real-time traffic controller, such would be in the spirit and scope of this disclosure, as would altering the presentation of the health monitor based upon a change of driver

30 status within the same confines.

Explanation of Figures

The system and method disclosed herein control the physical network interface card (NIC) in a computer 902. The firewall component (901) can be implemented as a kernel driver; and the interface to the display (904) can be a higher-level process or program communicating with the kernel driver.

5 Figures 1-3 show one embodiment that can be a standalone Trojan trapping system and/or implemented as a subprocess, which exposes and blocks Trojans within a larger firewall system. This embodiment processes the NIC's raw data packets 100 by dividing Internet-based packets into two categories: browser packets 120 and non-browser packets 102.

10 Figure 2 shows one example embodiment for processing browser packets. This embodiment first checks to see if the packet belongs to the subdomain of an open URL 201. Alternate embodiments might check to see if the packet belongs to the domain instead (e.g., alternate embodiments might approve cnn.com even if the URL itself is the subdomain sports.cnn.com).

15 If the packet does belong to the subdomain of an open URL 201 then the packet is allowed 210. If the packet does not belong to the subdomain of an open URL 201 then the packet is checked to see if belongs to any of the connections made by an open URL 202.

 If the packet does belong to a connection made by an open URL 202 then the
20 packet is allowed 210. If the packet does not belong to a connection made by an open URL 202 then the packet is blocked 203.

 Figure 3 shows one example embodiment for processing internet-based packets to and from non-browser apps 300. In this particular embodiment, the name of the maker of the app is obtained 301. Most software apps these days are digitally
25 signed. Such digital signatures contain the name of the maker of the app. This is one way of obtaining the name of the maker of the app. There are other methods well known in the art.

 The next step in this example embodiment is to obtain the name of the owner of the domain of the remote device to which the app is communicating 302. Various
30 means of obtaining this information include standard Whois lookups and/or proprietary databases (such as those published by numerous third parties).

The next step in this example embodiment is to determine whether the name of the app's maker matches the name of the domain owner 303. If the names do not match 303 then the packet is blocked 320; otherwise, the packet continues forward.

Figures 4 and 5 show two additional example embodiments for how browser
5 packets can be processed. There are numerous other possible alternative embodiments. Figure 4 is an example embodiment where the initial state is based upon the reputation of the web address. The user can then toggle the state thereafter. Figure 5 is an embodiment that does not use a reputation engine.

In the embodiment shown in Figure 4, browser packets 400 are processed by
10 first translating the remote IP address into its subdomain and domain 401. For example, an IP address might be translated into subdomain maps.google.com and domain google.com. Methods for mapping IP addresses into domains/subdomains have been disclosed in U.S. Patent No. 9,467,324. The system next checks to see if the subdomain matches an open URL 402, i.e., a web address that is open within the
15 browser app. A browser plugin can be used to transmit open and closed web addresses. Web addresses can also be obtained from HTTP/HTTPS interception, and/or remote web proxies.

If the subdomain does not match an open URL 402, the system then checks the subdomain to see if it belongs to any of the connections made by an open URL
20 403, and if it does belong then it is allowed but not displayed 404. The connections made by an open web address can be determined by a browser plugin using standard Web Extensions API calls, or for greater security, the web address connections can be downloaded from a local LAN server or remote WAN server which retrieves the web address connections from one or more trusted machines that accessed the same
25 web address and/or a statistical determination made from multiple user machines that accessed the same web address. Alternatively or adjunctively accessing web pages via a remote web proxy can allow correct web address connections to be obtained from and/or managed by the remote web proxy.

If the subdomain does not match an open web address and it is not from any
30 connection made by an open web address then the domain owner name is retrieved 420. The domain owner name can be obtained by a Whois lookup and/or from

proprietary database lookups (such as in cases where anonymous domain privacy services are returned in the Whois lookup). The domain name and its respective owner are displayed 421.

5 The packet is then checked to see if it is the first packet from the domain 422 (or alternatively from the subdomain). If it is the first packet received from the subdomain, then the subdomain's reputation is checked 423 to determine whether to allow or block it 424. If this is not the first packet from the domain/subdomain, then the current state is retrieved 440 in order to determine whether to allow or block it 441.

10 In the embodiment of the system shown in Figure 5, browser packets 500 are processed by first translating the remote IP address into its subdomain and domain 501. Then, the owner is retrieved 502. If the subdomain matches an open URL 503 then the owner and domain (and/or subdomain) are displayed and allowed 504.

15 If the subdomain does not match an open URL 503 then the system of this embodiment checks to see if the subdomain belongs to a connection made by an open URL 520. The same methods discussed above can be used to determine connections made by open web addresses. If the subdomain belongs to a connection made by an open URL 520 then it is allowed but not displayed 521; otherwise, the subdomain is blocked and displayed 540.

20 Figures 4 and 5 both illustrate examples of embodiments of the system in which the connections made by open web addresses can be allowed yet not displayed in order for Trojans that hijack browsers to easily be exposed and blocked. Any Trojan hijacking a browser will be displayed in these embodiments. Meanwhile, the plethora of extraneous web address connections will not be displayed so that the
25 Trojan's presence is readily seen by the user and blocked.

 Figures 6 and 7 illustrate additional example embodiments for processing non-browser data packets, i.e., data packets sent to and from a non-browser app. Numerous alternative embodiments are possible. The Figure 6 embodiment illustrates a manual implementation in which every new app/destination pair is
30 initially blocked. The user can then manually toggle the status thereafter. The Figure 7 embodiment illustrates an automated implementation that uses a

combination of Maker-Based Whitelisting and Consensus-Based Whitelisting to determine the initial state of each app/destination pair. The user can then toggle the status thereafter, if the user so chooses.

5 The Figure 6 embodiment begins by first examining whether the packet is either a DNS packet or local packet 601. If it is either of these types of data packets, then the packet is allowed but not displayed 604. Alternatively, the DNS packets can be processed in accordance with the methods disclosed in U.S. Patent No. 9,467,324 for greater security.

10 If the packet is neither a DNS packet nor a local packet 601 then the name of the application with which it is communicating is retrieved 602. Modern operating systems provide Transport Layer API calls to obtain such information. Next, the IP address of the remote machine is translated into a subdomain 603. Next, the owner of the subdomain is retrieved 620. The app and its owner are then displayed 621.

15 If this is the first packet from the app/destination pair 622 then the initial status is set to blocked 623. If this is not the first packet from the app/destination pair 622 then the current state is retrieved 640 in case the user previously has manually allowed the pair. If the current state is not allowed 641 then the pair remains blocked 623. If the current state is allowed 641 then the pair remains allowed 642. Note that the manual toggling of states is described in U.S. Patent No. 9,467,324, which is incorporated herein in its entirety by this reference.

20 The Figure 7 embodiment is the same as the Figure 6 embodiment until the App/Owner Name are displayed 721. Then, at this point, this embodiment uses Consensus-Based whitelisting to determine initial status.

25 The Figure 7 embodiment checks to see if the maker of the app matches the domain owner 722. If it does then the packet is initially allowed 723. If the makers of the app does not match the domain owner 722 then a consensus list of app/destination pairs is retrieved from a local LAN server and/or a remote WAN server 740. If the destination is in the consensus list 741 then the pair is initially allowed 723; otherwise, the pair is initially blocked 742.

30 Figure 8 illustrates an embodiment implementing the tri-level security of Automated Health Monitor, Maker-Based Whitelisting, and Consensus-Based

Whitelisting. As long as the Health Monitor is transitioning in color (e.g., grey → blue → grey → repeat), the Maker-Based Whitelisting and Consensus-Based Whitelisting can be trusted. Otherwise, the user is alerted to disconnect from the Internet to keep the computing device contents safe. Any continually changing
5 characteristic (e.g., color, shape, etc.) can be used in the Health Monitor to signify ongoing security.

Figure 9 illustrates the system and method's control of the physical components of a computing device. The user keyboard, mouse, touchscreen, etc., 900 are used to toggle state changes (see U.S. Patent No. 9,467,324 for alternative
10 embodiment implementations). The firewall itself 901 blocks/allows the data packets of the NIC 902 in accordance with the disclosure contained herein as well as the disclosure contained in U.S. Patent No. 9,467,324. The server 905 houses the connections made by various web addresses to be retrieved and used for Consensus-Based Whitelisting. The server can also house app/destination pairs for
15 embodiments that use Consensus-Based Whitelisting for non-browser apps.

Other Embodiments

It is to be understood that while the invention has been described in conjunction with the detailed description thereof, the foregoing description is intended to illustrate and not limit the scope of the invention, which is defined by
20 the scope of the appended claims. Other aspects, advantages, and modifications are within the scope of the following claims.

CLAIMS

What is claimed is:

- 5 1. A system for detecting and blocking Trojan malware and other malware, the system comprising:
- a computing device that is communicatively connected to a communications network;
- a browser software application being operated on the computing device;
- 10 an open URL identification process to communicate an identity of a web address, wherein the web address is an open web address if it is open in an address bar of the browser software application;
- an open connection tracking process that tracks at least one connection, if any such at least one connection is made, to a remote computing device made by the
- 15 open web address; and
- a firewall to intercept data packets sent to or from the browser software application, wherein the firewall blocks data packets that are not being sent to or from the open web address or to or from the at least one connection of the open web address, if any such at least one connection exists.
- 20 2. The system for detecting and blocking Trojan malware and other malware of claim 1, wherein the web address comprises a URL name, a path name of a URL, a domain name of a URL, or a subdomain of a URL; and wherein the identity comprises an owner name, a URL name, a path name of a URL, a domain
- 25 name of a URL, a subdomain of a URL, a derivative of one of the foregoing, or an alias representing one of the foregoing.
3. The system for detecting and blocking Trojan malware and other malware of claim 1, wherein the at least one connection, if any, made by the web
- 30 address is reported by the open connection tracking process as a URL name, a path name of a URL, a resource named by a URL, a domain name of a URL, or a

subdomain of a URL; and wherein the identity comprises an owner name, a URL name, a path name of a URL, a resource named by a URL, a domain name of a URL, a subdomain of a URL, a derivative of one of the foregoing, or an alias representing one of the foregoing.

5

4. The system for detecting and blocking Trojan malware and other malware of claim 1, wherein the open connection tracking process verifies the at least one connection made by the open web address, wherein the system further comprises at least one trusted computing device to identify at least one authentic
10 open connection of the open web address, and wherein the at least one trusted computing device reports an authenticity list comprising each at least one connection of the web address that is authentic.

5. The system for detecting and blocking Trojan malware and other
15 malware of claim 1, further comprising a connection analysis engine to statistically analyze the connections reported by multiple devices that access the same open web address to verify that the at least one connection is a trusted connection that the connection analysis engine assigns a trusted status or is not a trusted connection that the connection analysis engine assigns a distrusted status.

20

6. The system for detecting and blocking Trojan malware and other malware of claim 5, wherein the connection analysis engine is operated on the computing device or on a different computing device that is communicatively connected to the computing device.

25

7. The system for detecting and blocking Trojan malware and other malware of claim 1, wherein the firewall is configured to intercept and automatically block data packets sent to or from the browser software application that are not being sent to or from the open web address or to or from the at least one connection, if any,
30 of the open web address.

8. The system for detecting and blocking Trojan malware and other malware of claim 1, further comprising a reputation engine to allow or block the open web address; wherein the firewall blocks the open web address if the open web address does not meet a reputation threshold as determined by the reputation engine.

5

9. The system for detecting and blocking Trojan malware and other malware of claim 1, further comprising a reputation engine to allow or block the at least one connection, if any, made by the open web address; wherein the firewall blocks the at least one connection, if any, made by the open web address if the at least one connection made by the open web address does not meet a reputation threshold as determined by the reputation engine.

10. The system for detecting and blocking Trojan malware and other malware of claim 1, further comprising a reputation engine to allow or block at least one other connection to the browser software application, wherein the firewall blocks the at least one other connection if the at least one other connection does not meet a reputation threshold as determined by the reputation engine, and wherein the at least one other connection comprises one or more connections that are not the open web address or the at least one connection to the open web address.

20

11. The system for detecting and blocking Trojan malware and other malware of claim 1, further comprising a web address identity interface to display the identity of the open web address to a user; wherein the web address identity interface comprises at least one control element that permits the user to allow or block data packets sent to and from the open web address by selecting an allowed status or a blocked status for the open web address; and wherein the at least one connection, if any, of the open web address is also assigned the blocked status when the user operates the at least one control element of the web address identity interface to block the open web address, and wherein the firewall blocks data packets sent to and from the at least one connection.

30

12. The system for detecting and blocking Trojan malware and other malware of claim 11, wherein the at least one connection of the open web address is allowed, and not blocked, when the open web address is blocked but a second open web address that is allowed shares the same at least one connection.

5

13. The system for detecting and blocking Trojan malware and other malware of claim 11, wherein a connection interface displays the at least one connection of the open web address, wherein the connection interface comprises at least one control element that permits the user to allow or block the at least one
10 connection, and wherein the firewall blocks data packets sent to and from the at least one connection if the blocked status is selected for the at least one connection; wherein the connection interface is the web address identity interface or a different interface and wherein the at least one control element of the connection interface is the at least one control element of the web address identity interface or a different at
15 least one control element.

14. The system for detecting and blocking Trojan malware and other malware of claim 11, wherein the interface displays the identity of the web address of automatically blocked data packets, and wherein the blocked status of the web
20 address of the automatically blocked data packets is overridable by at least one option selected from the group consisting of:

the user using the at least one control element of the interface to manually select the allowed status for the web address to override the blocked status of the web address;

25 an override process overriding the blocked status of the web address by comparison to a whitelist that informs the override process that the data packets are not being sent to or from the open web address or its at least one connection; and

a reputation engine overriding the blocked status of the web address when the data packets are not being sent to or from the open web address or its at least one
30 connection when a reputation threshold is met or exceeded as determined by the reputation engine.

15. The system for detecting and blocking Trojan malware and other malware of claim 1, wherein the open web address is a user-opened web address opened by the user and wherein the system further comprises an interface to display
5 the identity of the user-opened web address to a user, wherein the at least one connection comprises at least one connection to a remote computing device made by the user-opened web address, wherein the interface does not display the at least one connection made by the user-opened web address but does allow data packets sent to or from that at least one connection, and wherein the interface displays and
10 the firewall blocks data packets that are not sent to or received from the browser software application by the user-opened web address or its at least one connection.

16. The system for detecting and blocking Trojan malware and other malware of claim 15, further comprising an override process to override the blocked
15 status of the data packets that are not sent to or received from the browser software application by the user-opened web address or its at least one connection, wherein the override process compares a web address or identity of the blocked data packets to a whitelist that informs the override process that the blocked status of the blocked data packets should be overridden, and wherein the override process displays the
20 blocked data packets and changes their blocked status to allowed status if the web address or identity of the blocked data packets is located in the whitelist.

17. The system for detecting and blocking Trojan malware and other malware of claim 1, further comprising a web traffic analysis system to determine
25 the identity of the open web address, the identity of the at least one connection, if any, of the open web address, or the identity of both; and wherein the web traffic analysis system comprises a browser plugin, a local man-in-the-middle http/https interception, a remote man-in-the-middle http/https interception, or a web proxy.

30 18. The system for detecting and blocking Trojan malware and other malware of claim 1, further comprising an automated health monitor engine that

periodically checks a status of a kernel driver of the system to ensure that the kernel driver is operational.

19. The system for detecting and blocking Trojan malware and other
5 malware of claim 18, wherein the automated health monitor engine generates an alert if the kernel driver is disabled.

20. The system for detecting and blocking Trojan malware and other
malware of claim 18, further comprising an interface comprising a continuously
10 changing visual display, wherein a continuous change in appearance of the continuously changing visual display represents that the automated health monitor engine is checking the kernel driver, and wherein the continuously changing visual display ceases continuously changing in appearance if the automated health monitor engine is unable to perform the check of the status of the kernel driver.

15

21. A system for detecting and blocking Trojan malware and other malware, the system comprising:

a computing device that is communicatively connected to a communications network;

20 a browser software application being operated on the computing device;

a remote web proxy that replaces at least one connection made by an open web address, if any, with an alias based on a domain name of the web proxy or another domain name; and

25 a firewall that blocks all data packets not sent to or received from the web proxy by the browser software application;

wherein the web address is an open web address if it is open in the web proxy.

22. The system for detecting and blocking Trojan malware and other malware of claim 21, wherein the web proxy reports the open web address to the
30 firewall.

23. The system for detecting and blocking Trojan malware and other malware of claim 22, wherein a firewall interface displays the open web address.

24. The system for detecting and blocking Trojan malware and other malware of claim 22, wherein the firewall interface permits a user to select an allowed status or a blocked status for the web address, wherein the firewall communicates the web address to the web proxy when a blocked status is selected for the web address, and wherein the web proxy subsequently blocks data packets sent to or received from the web address.

10

25. The system for detecting and blocking Trojan malware and other malware of claim 21, wherein data packets not sent to or received from the web proxy by the browser software application are initially automatically blocked except data packets sent to or received from the browser software application by a web address or IP address that appears on a whitelist.

15

26. A system for detecting and blocking Trojan malware and other malware, the system comprising:

a computing device that is communicatively connected to a communications network;

20

at least one non-browser software application operated on the computing device;

a maker identification process to identify a maker of the at least one non-browser software application;

25

an identity verification process to determine an owner of at least one remote host address sending data packets to or receiving data packets from each at least one non-browser software application, wherein the owner of each at least one remote host address as determined by the identity verification process and the maker of each at least one non-browser software application are compared to determine whether they are the same or different; and

30

a firewall to block data packets sent to and from the at least one non-browser software application by each at least one remote host address when the owner of each at least one remote host address and the maker of each at least one non-browser software application are different.

5

27. The system for detecting and blocking Trojan malware and other malware of claim 26, wherein a mismatched application/remote host address communication pair comprises one at least one remote host address having an owner that is different from the maker of one at least one non-browser software application;
10 and wherein the identification of a mismatched application/remote host address communication pair results in the firewall blocking data packets transmitted between the mismatched application/remote host address pair.

28. The system for detecting and blocking Trojan malware and other
15 malware of claim 26, further comprising a reputation engine that initially blocks data packets from the at least one remote host address when a reputation of the at least one remote host address is determined by the reputation engine to not meet or exceed a reputation threshold.

29. The system for detecting and blocking Trojan malware and other
malware of claim 26, further comprising a first interface to display a name of each
at least one non-browser software application, the maker of each at least one non-
browser software application, each at least one remote host address with which each
at least one non-browser application attempts to communicate, and the owner of each
25 at least one remote host address; and a second interface that permits a user to
selectively allow or block traffic sent to and received from each at least one non-
browser software application by each at least one remote host address; wherein the
second interface is the first interface or a different interface.

30. The system for detecting and blocking Trojan malware and other
malware of claim 26, wherein each non-browser software application of the at least

one non-browser software application that communicates with a remote host address of the at least one web address is an application/remote host address communication pair; wherein the system comprises a process to initially block all application/remote host address communication pairs; wherein the system further comprises a first
5 interface to display a name of each non-browser software application, the maker of each non-browser software application, each web address with which each non-browser software application communicates, and the owner of each such web address; and wherein the system further comprises a second interface that permits a user to selectively allow or block traffic sent to and received from each at least one
10 non-browser software application by each at least one web address; wherein the second interface is the first interface or a different interface.

31. The system for detecting and blocking Trojan malware and other malware of claim 26, wherein each non-browser software application of the at least
15 one non-browser software application that communicates with a remote host address of the at least one remote host address is an application/remote host address communication pair; wherein the system comprises a process to initially block traffic between all application/remote host address communication pairs for a temporary period of time; wherein the system further comprises a first interface to display a
20 name of each non-browser software application, the maker of each non-browser software application, each remote host address with which each non-browser software application communicates, and the owner of each such remote host address; and wherein the system further comprises a second interface that permits a user to selectively quarantine one or more application/remote host address communication
25 pairs of all the application/remote host address communication pairs prior to expiration of the temporary period of time; wherein the second interface is the first interface or a different interface.

32. The system for detecting and blocking Trojan malware and other
30 malware of claim 26, wherein legitimate DNS data packets and local packets and traffic to digital certificate authorities are automatically allowed.

33. A system for detecting and blocking Trojan malware and other malware, the system comprising:

5 a computing device that is communicatively connected to a communications network;

at least one non-browser software application being operated on the computing device;

10 an identity verification process to determine an owner of at least one remote host address sending data packets to or receiving data packets from each at least one non-browser software application;

an interface to communicate both an identity of the at least one non-browser software application and the owner of the at least one remote host address with which the at least one non-browser software application is communicating; and

15 a firewall to block data packets transmitted between at least one application/destination pair, wherein the at least one application/destination pair comprises one of the at least one non-browser applications communicating with one of the at least one remote host addresses that is the destination of the at least one application/destination pair.

20 34. The system for detecting and blocking Trojan malware and other malware of claim 34, further comprising an interface to permit a user to selectively allow or block one or more of the at least one application/destination pairs.

25 35. The system for detecting and blocking Trojan malware and other malware of claim 34, wherein the firewall automatically blocks communication between each newly encountered application/destination pair of the at least one application/destination pair; and wherein the interface permits the user to toggle a current status of each at least one application/destination pair between an allowed status and a blocked status.

30

36. The system for detecting and blocking Trojan malware and other malware of claim 34, wherein the firewall automatically blocks communication between each newly encountered application/destination pair of the at least one application/destination pair unless the newly encountered application/destination pair, the destination, or both are whitelisted, and wherein the interface permits the user to toggle a current status of each at least one application/destination pair between an allowed status and a blocked status.

37. The system for detecting and blocking Trojan malware and other malware of claim 34, wherein the firewall automatically blocks communication between each newly encountered application/destination pair of the at least one application/destination pair unless a destination reputation of the destination meets or exceeds a destination reputation threshold, wherein the system further comprises a destination reputation engine to determine the destination reputation of the destination, and wherein the interface permits the user to toggle a current status of each at least one application/destination pair between an allowed status and a blocked status.

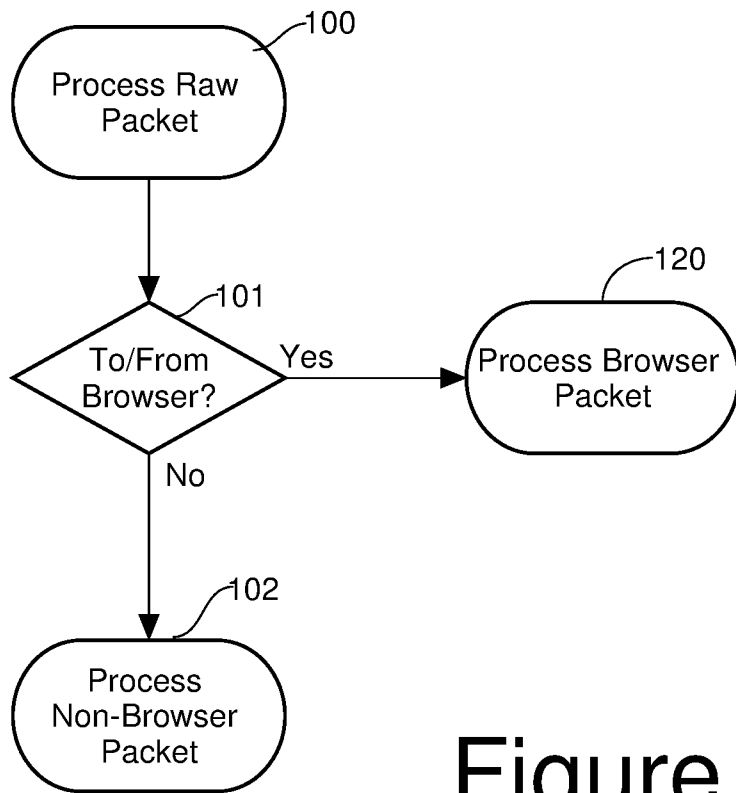


Figure 1

Figure 2

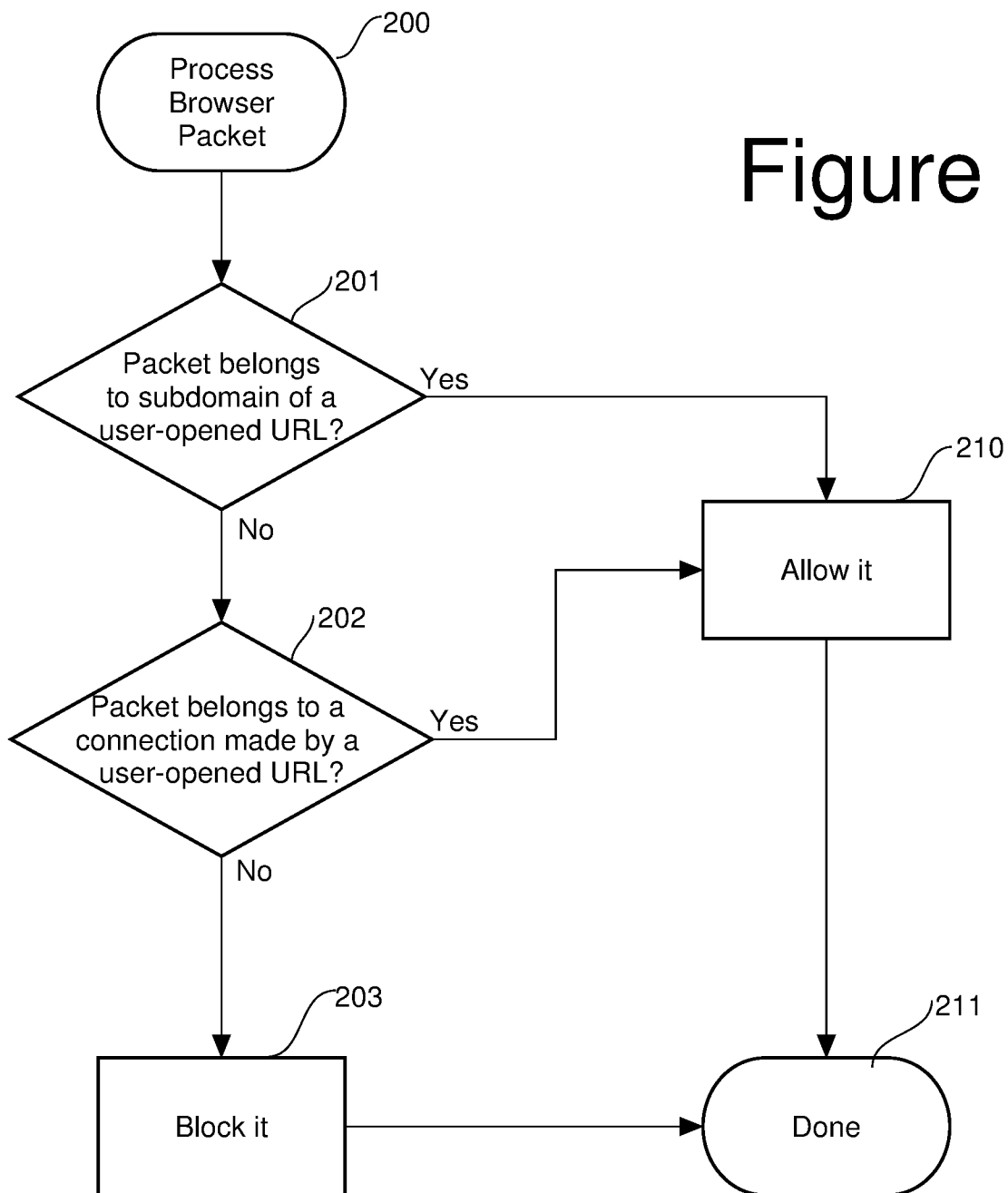
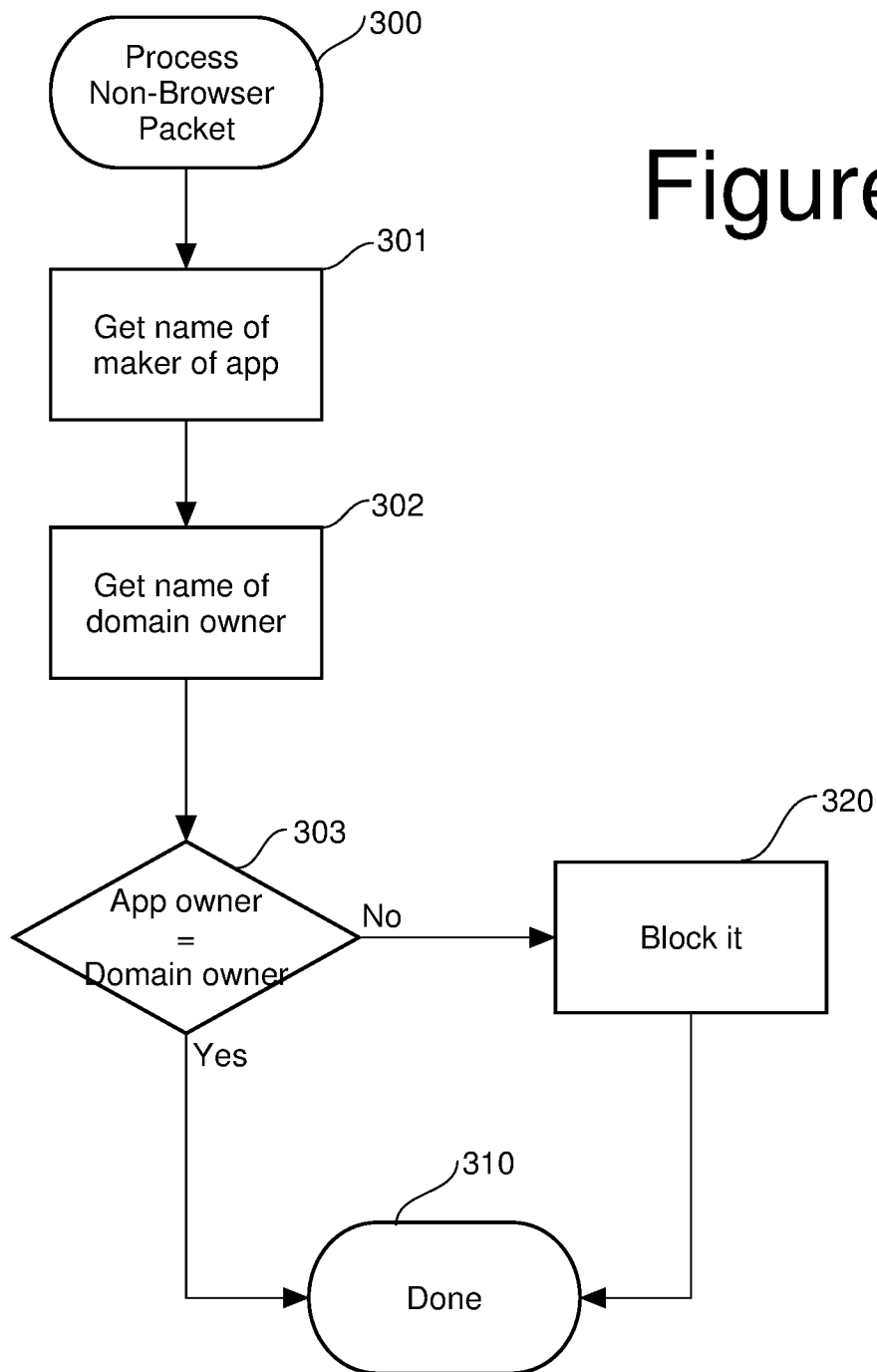


Figure 3



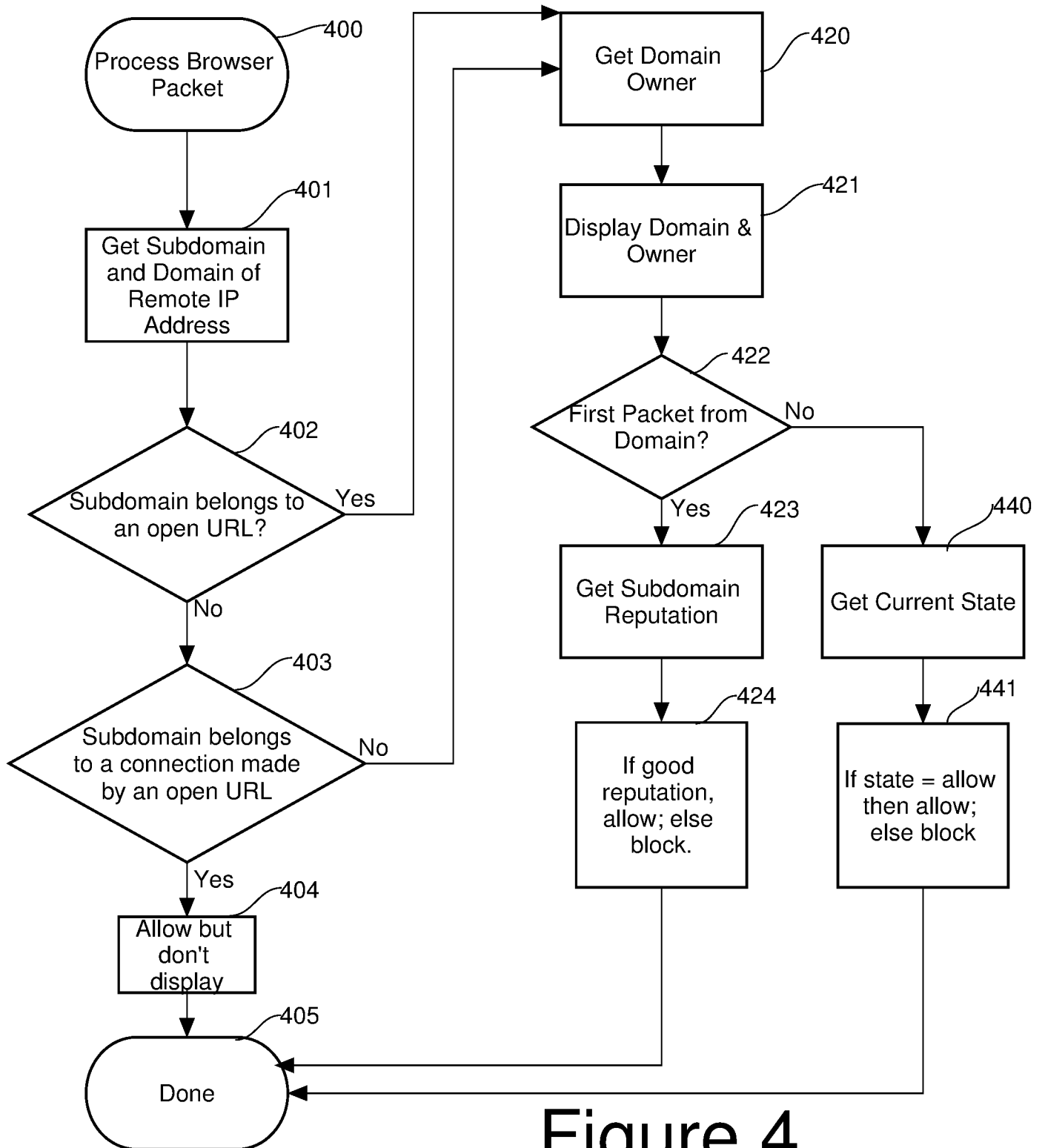


Figure 4

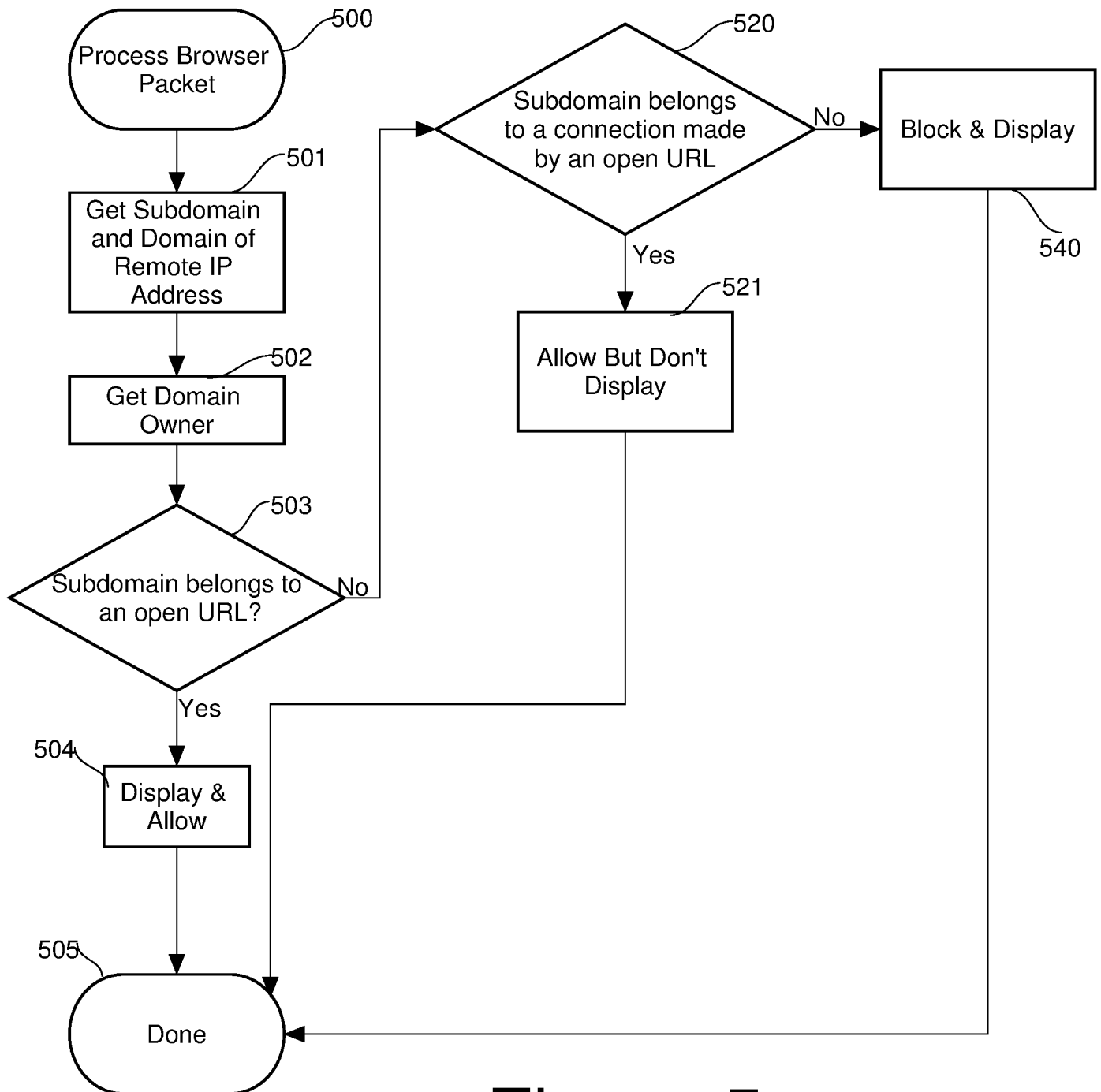


Figure 5

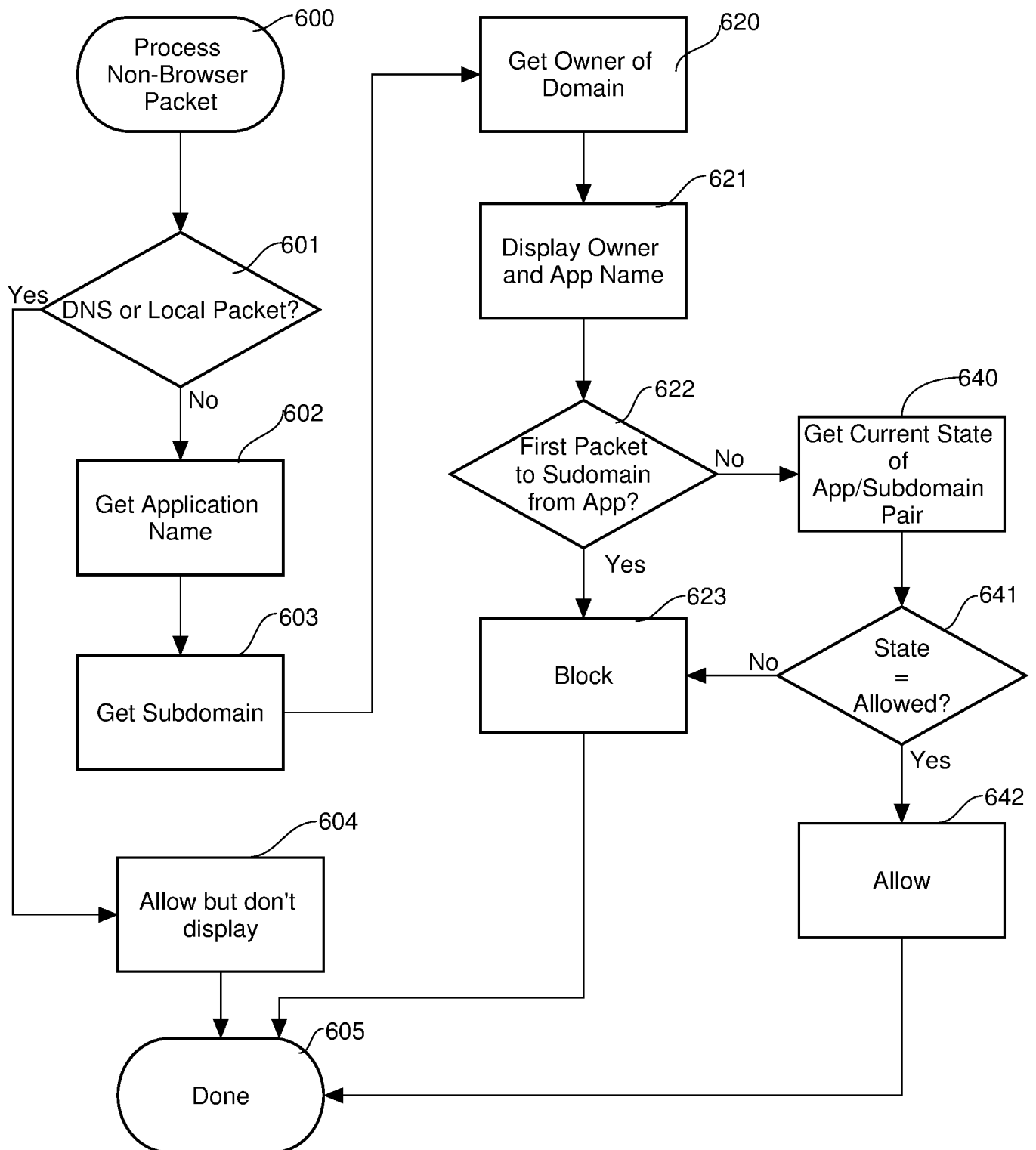


Figure 6

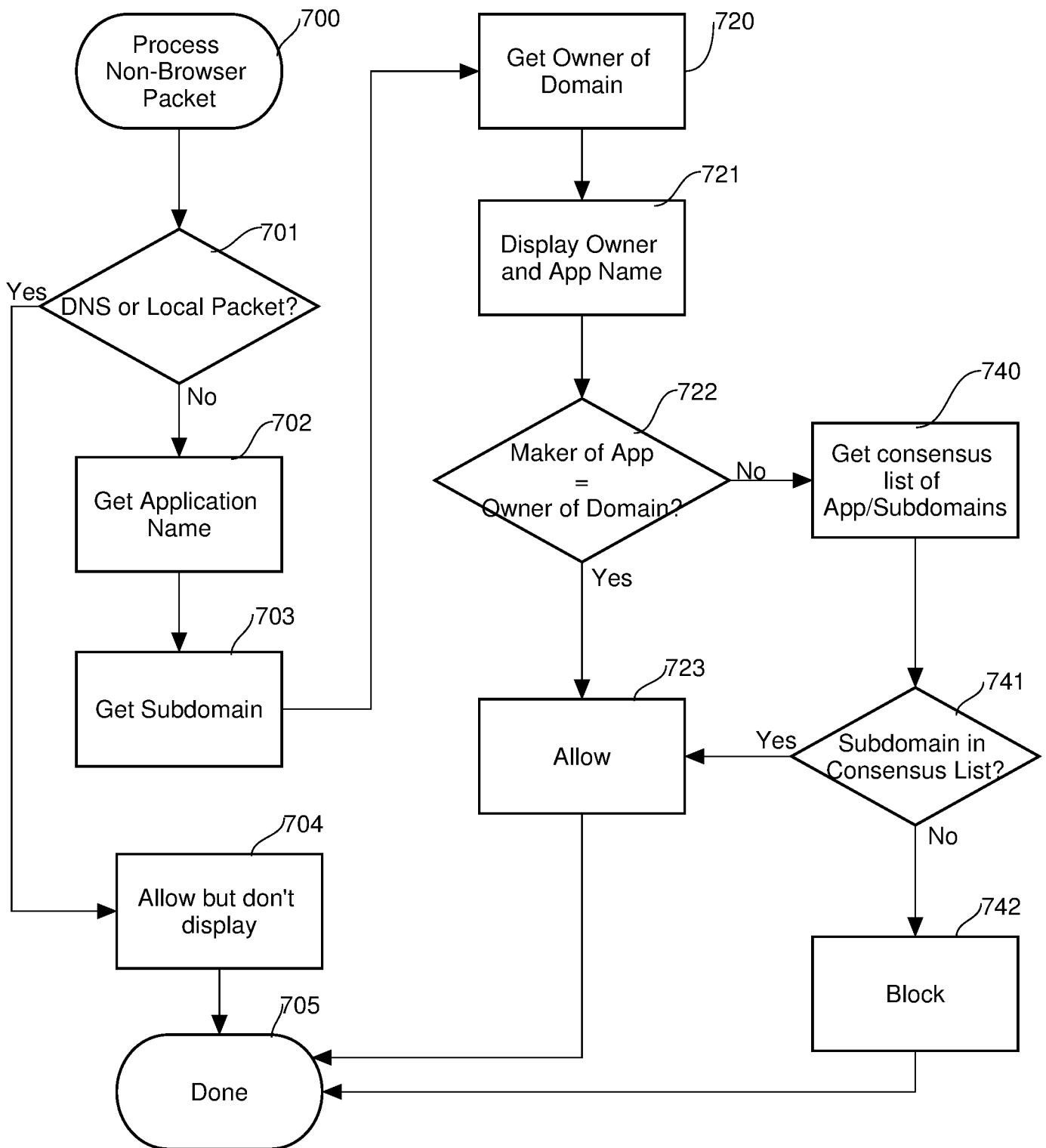


Figure 7

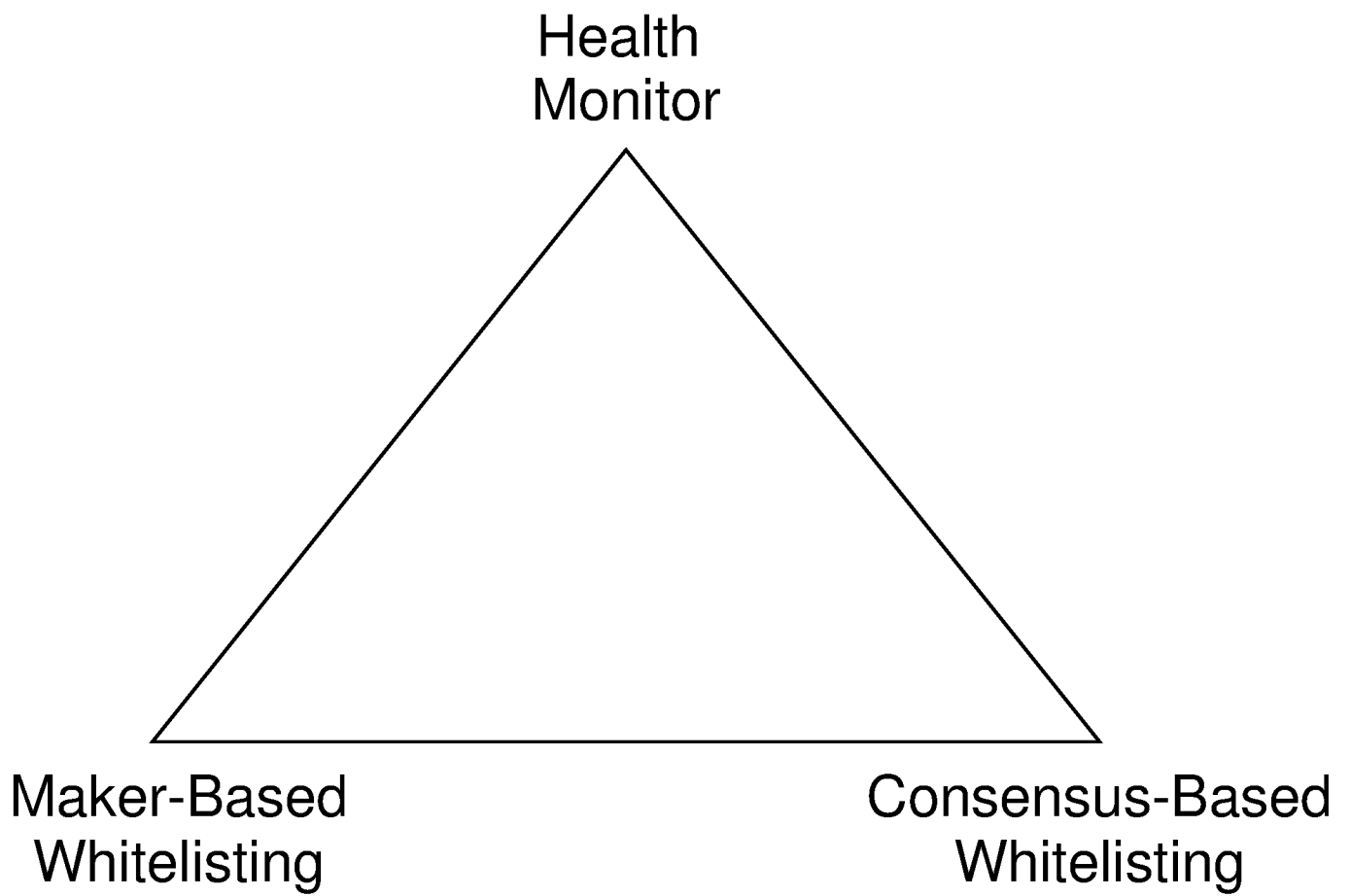


Figure 8

Figure 9

