

DOMAIN NAME SYSTEM (DNS) SECURITY

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 63/483,628 filed February 7, 2023, entitled, "Domain Name System (DNS) Security," which is incorporated herein by reference in its entirety.

COPYRIGHT STATEMENT

[0002] A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

FIELD

[0003] The present disclosure relates, in general, to methods, systems, and apparatuses for implementing domain name system ("DNS") functionality, and, more particularly, to methods, systems, and apparatuses for implementing improvement to DNS security, by requiring DNS queries to be reattempted.

BACKGROUND

[0004] User datagram protocol ("UDP") -based domain name system ("DNS") requests or packets can be easily source-spoofed, leading to its use in malicious network attacks as distributed denial of service ("DDoS") traffic. In fact, many well-orchestrated UDP-based DNS attacks can be virtually indistinguishable from normal DNS traffic.

[0005] It is with respect to this general technical environment to which aspects of the present disclosure are directed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] A further understanding of the nature and advantages of particular embodiments may be realized by reference to the remaining portions of the specification and the drawings, in which like reference numerals are used to refer to similar components. In some instances, a sub-label is associated with a reference numeral to denote one of multiple similar components. When reference is made to a reference numeral without specification to an existing sub-label, it is intended to refer to all such multiple similar components. For denoting a plurality of components, the suffixes "a" through "n," where n denotes any suitable integer number, and may be either the same or different from the suffix "n" for other components in the same or different figures. For example, for component #1 105a-105n, the integer value of n in 105n may be the same or different from the integer value of n in 110n for component #2 110a-110n, and so on.

[0007] Fig. 1 is a schematic diagram illustrating a system for implementing improvement to DNS security, in accordance with various embodiments.

[0008] Fig. 2 is a schematic diagram illustrating a non-limiting example of a grant access condition when implementing improvement to DNS security, in accordance with various embodiments.

[0009] Fig. 3 is a schematic diagram illustrating a non-limiting example of a deny access condition when implementing improvement to DNS security, in accordance with various embodiments.

[0010] Figs. 4A-4F are flow diagrams illustrating a method for implementing improvement to DNS security, in accordance with various embodiments.

[0011] Fig. 5 is a block diagram illustrating an exemplary computer or system hardware architecture, in accordance with various embodiments.

DETAILED DESCRIPTION OF CERTAIN EMBODIMENTS

[0012] Overview

[0013] Various embodiments provide tools and techniques for implementing domain name system ("DNS") functionality, and, more particularly, to methods, systems, and apparatuses for implementing improvement to DNS security, by requiring DNS queries to be reattempted.

[0014] In various embodiments, a computing system may receive a first user datagram protocol ("UDP") -based DNS request, the first UDP-based DNS request including a source address and a query for a destination DNS record associated with at least one of a destination device in a network, a destination entity associated with the destination device, or a destination domain associated with the destination device or the destination entity. In response to receiving the first UDP-based DNS request, the computing system may send a UDP-based response message to the source address. In some instances, the response message may include an empty payload portion and a header portion containing a truncate ("TC") flag that is set. When a first transmission control protocol ("TCP") -based DNS request corresponding to the first UDP-based DNS request is received from the source address within a first predetermined period, the computing system may send, to the source address, a TCP-based response message comprising an answer to the query for the destination DNS record. When a TCP-based DNS request corresponding to the first UDP-based DNS request is not received from the source address within the first predetermined period, the computing system may cause one or more second UDP-based DNS requests from the source address to be blocked. In some cases, the one or more second UDP-based DNS requests may include the first UDP-based DNS request.

[0015] In this manner, the various embodiments leverage the features of the truncate flag as a novel feature for improving DNS security. The truncate flag is conventionally used in a UDP response that exceeds 512 bytes in payload size (and thus has its payload truncated) to request or indicate resending of a UDP message (to which the UDP response is responding to) as a TCP message, so that a corresponding TCP response (which is not bound by the 512 byte payload limitation) may be sent. For example, a query from a new client's source IP address (or source port) may be received and an empty response may be returned with the truncated flag set. The client's behavior would be to reattempt the query via TCP. This retry indicates the source IP address (or source port) will be temporarily trusted for UDP traffic. Additional UDP attempts, while being evaluated, may be fed into an AI/ML system to determine whether or not to ban or block the source IP address (or source port), and for how long. Because IP addresses are

ephemeral, the AI/ML system may provide additional tracking to ensure the client's behavior has not soured and/or to remove any imposed bans.

[0016] These and other aspects of the improvement to DNS security are described in greater detail with respect to the figures.

[0017] The following detailed description illustrates a few exemplary embodiments in further detail to enable one of skill in the art to practice such embodiments. The described examples are provided for illustrative purposes and are not intended to limit the scope of the invention.

[0018] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the described embodiments. It will be apparent to one skilled in the art, however, that other embodiments of the present invention may be practiced without some of these specific details. In other instances, certain structures and devices are shown in block diagram form. Several embodiments are described herein, and while various features are ascribed to different embodiments, it should be appreciated that the features described with respect to one embodiment may be incorporated with other embodiments as well. By the same token, however, no single feature or features of any described embodiment should be considered essential to every embodiment of the invention, as other embodiments of the invention may omit such features.

[0019] Unless otherwise indicated, all numbers used herein to express quantities, dimensions, and so forth used should be understood as being modified in all instances by the term "about." In this application, the use of the singular includes the plural unless specifically stated otherwise, and use of the terms "and" and "or" means "and/or" unless otherwise indicated. Moreover, the use of the term "including," as well as other forms, such as "includes" and "included," should be considered non-exclusive. Also, terms such as "element" or "component" encompass both elements and components comprising one unit and elements and components that comprise more than one unit, unless specifically stated otherwise.

[0020] In an aspect, a method may comprise receiving, by a computing system of a domain name system ("DNS"), a first user datagram protocol ("UDP") -based DNS request, the first UDP-based DNS request comprising a source address and a query for a destination DNS record associated with at least one of a destination device in a network, a destination entity associated with the destination device, or a destination domain associated with the destination device or the destination entity; in response to receiving the first UDP-based DNS request, sending, by the

computing system, a UDP-based response message to the source address, the UDP-based response message comprising an empty payload portion and a header portion containing a truncate ("TC") flag that is set; when a first transmission control protocol ("TCP") -based DNS request corresponding to the first UDP-based DNS request is received from the source address within a first predetermined period, sending, by the computing system and to the source address, a TCP-based response message comprising an answer to the query for the destination DNS record; and when a TCP-based DNS request corresponding to the first UDP-based DNS request is not received from the source address within the first predetermined period, causing, by the computing system, UDP-based DNS requests from the source address to be blocked.

[0021] In some embodiments, the computing system may comprise at least one of a DNS resolver, a DNS recursive resolver ("recursor"), a DNS root nameserver, a top level domain ("TLD") nameserver, an authoritative nameserver, a threat intelligence system, a threat mitigation system, a server, an artificial intelligence ("AI") system, a machine learning ("ML") system, a virtual machine ("VM"), or software running on the computing system, and/or the like. In some cases, the response message may comprise an empty payload portion. In some instances, the source address may comprise at least one of a source port or a source IP address.

[0022] According to some embodiments, sending the TCP-based response message may comprise one of: sending, by the computing system and to the source address, the destination DNS record based on the first UDP-based DNS request, the destination DNS record being obtained from at least one of a cache of the computing system or an authoritative nameserver search; or sending, by the computing system and to the source address, a message indicating that the destination DNS record was not found.

[0023] In some embodiments, the method may further comprise after receiving the first TCP-based DNS request corresponding to the first UDP-based DNS request, updating, by the computing system, one or more rules of at least one access control list ("ACL") to grant access to the network to the source address.

[0024] According to some embodiments, the method may further comprise, after receiving the first TCP-based DNS request corresponding to the first UDP-based DNS request, performing at least one of: causing, by the computing system, one or more second UDP-based DNS requests that are received by the computing system from the source address to be processed; or causing, by the computing system, a threat intelligence system to communicate to each of a plurality of

DNS resolvers to allow one or more third UDP-based DNS requests that are received by the plurality of DNS resolvers from the source address to be processed.

[0025] In some embodiments, the method may further comprise causing, by the computing system, one or more fourth UDP-based DNS requests that are received, within a second predetermined period after receiving the first TCP-based DNS request, from the source address to be processed.

[0026] In some examples, the method may further comprise analyzing, by the computing system and using a machine learning model, one or more fifth UDP-based DNS requests that are received after the second predetermined period, from the source address, to determine whether or not to block the one or more fifth UDP-based DNS requests; and performing, by the computing system, one or more DNS tasks based on the analysis.

[0027] In some embodiments, causing the UDP-based DNS requests from the source address to be blocked may comprise at least one of: updating, by the computing system, one or more rules of at least one access control list ("ACL") to deny access to the network to the source address; dropping, by the computing system, all DNS requests from the source address; or filtering, by the computing system, all DNS requests from the source address; and/or the like. In some cases, causing the first UDP-based DNS request from the source address to be blocked may comprise causing, by the computing system, all DNS requests that are received from the source address over a third predetermined period to be blocked.

[0028] According to some embodiments, the method may further comprise dropping, by the computing system, any one or more sixth UDP-based DNS requests that are received from the source address after receiving the first UDP-based DNS request and before receiving the first TCP-based DNS request.

[0029] Merely by way of example, in some cases, at least one of a query, a query type, a query source, or a mail exchange ("MX") record of the first UDP-based DNS request may be the same as corresponding at least one of a query, a query type, a query source, or a MX record of the first TCP-based DNS request.

[0030] In some embodiments, the method may further comprise determining, by the computing system, whether the first UDP-based DNS request is a legitimate DNS request, by performing at least one of: determining, by the computing system, that the first UDP-based DNS request is among a number of UDP-based DNS requests that exceeds a predetermined number of

requests within a fourth predetermined period and that are received from the source address; or determining, by the computing system, that an ACL already indicates that the source address should be denied access to the network; and/or the like.

[0031] In another aspect, a system may comprise a computing system of a domain name system ("DNS"). The computing system may comprise at least one first processor and a first non-transitory computer readable medium communicatively coupled to the at least one first processor. The first non-transitory computer readable medium may have stored thereon computer software comprising a first set of instructions that, when executed by the at least one first processor, causes the computing system to: receive a first user datagram protocol ("UDP") -based DNS request, the first UDP-based DNS request comprising a source address and a query for a destination DNS record associated with at least one of a destination device in a network, a destination entity associated with the destination device, or a destination domain associated with the destination device or the destination entity; in response to receiving the first UDP-based DNS request, send a UDP-based response message to the source address, the UDP-based response message comprising an empty payload portion and a header portion containing truncate ("TC") flag that is set; when a first transmission control protocol ("TCP") -based DNS request corresponding to the first UDP-based DNS request is received from the source address within a first predetermined period, send, to the source address, a TCP-based response message comprising an answer to the query for the destination DNS record; and when a TCP-based DNS request corresponding to the first UDP-based DNS request is not received from the source address within the first predetermined period, cause UDP-based DNS requests from the source address to be blocked.

[0032] In some embodiments, the computing system may comprise at least one of a DNS resolver, a DNS recursive resolver ("recursor"), a DNS root nameserver, a top level domain ("TLD") nameserver, an authoritative nameserver, a threat intelligence system, a threat mitigation system, a server, an artificial intelligence ("AI") system, a machine learning ("ML") system, a virtual machine ("VM"), or software running on the computing system, and/or the like. In some instances, the response message may comprise an empty payload portion. In some instances, the source address may comprise at least one of a source port or a source IP address.

[0033] According to some embodiments, causing the UDP-based DNS requests from the source address to be blocked may comprise at least one of: updating one or more rules of at least

one access control list ("ACL") to deny access to the network to the first source address; dropping all DNS requests from the first source address; or filtering all DNS requests from the first source address; and/or the like.

[0034] In yet another aspect, a method may comprise receiving, by a computing system of a domain name system ("DNS"), a first user datagram protocol ("UDP") -based DNS request, the first UDP-based DNS request comprising a source address and a query for a destination DNS record associated with at least one of a destination device in a network, a destination entity associated with the destination device, or a destination domain associated with the destination device or the destination entity; in response to receiving the first UDP-based DNS request, sending, by the computing system, a UDP-based response message to the source address, the UDP-based response message comprising an empty payload portion and a header portion containing a truncate ("TC") flag that is set; and when a transmission control protocol ("TCP") -based DNS request corresponding to the first UDP-based DNS request is not received from the source address within a first predetermined period, causing, by the computing system, one or more second UDP-based DNS requests from the source address to be blocked, the one or more second UDP-based DNS requests comprising the first UDP-based DNS request. The method may further comprise, when a first TCP-based DNS request corresponding to the first UDP-based DNS request is received from the source address within the first predetermined period, allowing, by the computing system and within a second predetermined period after receiving the first TCP-based DNS request, at least the first UDP-based DNS request from the source address to be processed, by sending, by the computing system and to the source address, at least a TCP-based response message comprising an answer to the query for the destination DNS record; receiving, by the computing system and from the source address, one or more third UDP-based DNS requests, after the second predetermined period; analyzing, by the computing system and using a machine learning model, the one or more third UDP-based DNS requests, to determine whether or not to block the one or more third UDP-based DNS requests; and performing, by the computing system, one or more DNS tasks based on the analysis.

[0035] According to some embodiments, the computing system may comprise at least one of a DNS resolver, a DNS recursive resolver ("recursor"), a DNS root nameserver, a top level domain ("TLD") nameserver, an authoritative nameserver, a threat intelligence system, a threat

mitigation system, a server, an artificial intelligence ("AI") system, a machine learning ("ML") system, a virtual machine ("VM"), or software running on the computing system, and/or the like.

[0036] In some embodiments, causing the one or more second UDP-based DNS requests from the source address to be blocked may comprise at least one of: updating, by the computing system, one or more rules of at least one access control list ("ACL") to deny access to the network to the source address; dropping, by the computing system, all DNS requests from the source address; or filtering, by the computing system, all DNS requests from the source address; and/or the like.

[0037] Various modifications and additions can be made to the embodiments discussed without departing from the scope of the invention. For example, while the embodiments described above refer to particular features, the scope of this invention also includes embodiments having different combination of features and embodiments that do not include all of the above-described features.

[0038] **Specific Exemplary Embodiments**

[0039] We now turn to the embodiments as illustrated by the drawings. Figs. 1-5 illustrate some of the features of the method, system, and apparatus for implementing domain name system ("DNS") functionality, and, more particularly, to methods, systems, and apparatuses for implementing improvement to DNS security, by requiring DNS queries to be reattempted, as referred to above. The methods, systems, and apparatuses illustrated by Figs. 1-5 refer to examples of different embodiments that include various components and steps, which can be considered alternatives or which can be used in conjunction with one another in the various embodiments. The description of the illustrated methods, systems, and apparatuses shown in Figs. 1-5 is provided for purposes of illustration and should not be considered to limit the scope of the different embodiments.

[0040] With reference to the figures, Fig. 1 is a schematic diagram illustrating a system 100 for implementing improvement to DNS security, in accordance with various embodiments.

[0041] In the non-limiting embodiment of Fig. 1, system 100 may include, without limitation, at least one of computing system 105, one or more DNS resolvers 110a-110n (collectively, "DNS resolvers 110" or the like), one or more access control lists ("ACLs") or filters 115a and/or 115b (collectively, "ACL or filter 115" or the like), cache 120, threat intelligence system 125, artificial intelligence ("AI")/machine learning ("ML") system 130,

authoritative name server or nameserver 135, database(s) or DNS database(s) 140, one or more networks 145a-145n (collectively, "network(s) 145" or the like), one or more first client devices 150a associated with a first entity 155a, or one or more second client devices 150b associated with a second entity 155b, and/or the like. According to some embodiments, ACL or filter 115a and cache 120 may be integrated with or within each of at least one DNS resolver 110 among the one or more DNS resolvers 110a-110n, while ACL or filter 115b may be disposed in a network(s) 145 and external to any DNS resolver 110a-110n, in some cases as part of a router ACL or filter or as part of a third party ACL or filter (as shown, e.g., in Fig. 2 or 3, or the like). Each ACL or filter 115 may indicate which source addresses (e.g., source ports and/or source Internet Protocol ("IP") addresses, or the like) are granted or denied access to the network(s) (in this case, network(s) 145, or the like, and thus determine to which source addresses the DNS system will return DNS responses (e.g., destination DNS records, or the like) to DNS queries, or the like. Herein, destination DNS records may include, but is not limited to, at least one of one or more destination A records (each including an IPv4 address of a destination domain, or the like), one or more destination AAAA records (each including an IPv6 address of a destination domain, or the like), one or more destination message exchange ("MX") records (each including routing information to a destination mail server, or the like), one or more destination nameserver ("NS") records (each including authoritative DNS server information for a destination domain), or one or more other DNS records or DNS record types, or the like. In some instances, AI/ML system 130 may be integrated with threat intelligence system 125. Alternatively, AI/ML system 130 may be separate from, yet communicatively coupled with, threat intelligence system 125 (not shown). In some embodiments, each of computing system 105, each DNS resolver among DNS resolvers 110a-110n, threat intelligence system 125, and authoritative name server 135 and corresponding database(s) 140 may be disposed within a network(s) 145 among the one or more networks 145a-145n. In some cases, database(s) 140 may be a distributed database spanning a number of networks among the one or more networks 145a-145n.

[0042] In some embodiments, the computing system 105 may include, without limitation, at least one of a DNS resolver (e.g., DNS resolver 110 among the one or more DNS resolvers 110a-110n, or the like), a DNS recursive resolver ("recursor," which may be configured to query other DNS resolvers 110 or DNS servers for the destination DNS record; herein, each DNS resolver 110 represents either a DNS resolver or a DNS recursor, or the like), a DNS root

nameserver, a top level domain ("TLD") nameserver, an authoritative nameserver, a threat intelligence system (e.g., threat intelligence system 125, or the like), a threat mitigation system (similar to threat intelligence system 125, or the like), a server, an AI system or a ML system (e.g., AI/ML system 130, or the like), a virtual machine ("VM"), or software (e.g., daemon, script, or other software, or the like) running on the computing system 105, and/or the like. In some cases, the authoritative nameserver 135 may include at least one of the DNS root nameserver, the TLD nameserver, and/or the authoritative nameserver, and/or the like. In some instances, the one or more client devices 150a or 150b may each include, but is not limited to, one of a desktop computer, a laptop computer, a tablet computer, a smart phone, a mobile phone, a server, a cloud computing system, a distributed computing system, or any suitable device capable of communicating with a DNS resolver (e.g., DNS resolver 110 or 110a-110n, or the like) and/or other computing system (e.g., computing system 105, or the like) via a web-based portal, a web browser, an application programming interface ("API"), a server, a software application ("app"), or any other suitable communications interface, or the like, over network(s) 145a-145n. In some cases, the entity 155a or 155b may include, without limitation, an individual, a group of individuals, a company, a group of companies, a (foreign) government actor, a (foreign) government agency, or an alliance of (foreign) governmental actors, agencies, and/or entities, or the like.

[0043] According to some embodiments, network(s) 145a-145n may each include, without limitation, one of a local area network ("LAN"), including, without limitation, a fiber network, an Ethernet network, a Token-Ring™ network, and/or the like; a wide-area network ("WAN"); a wireless wide area network ("WWAN"); a virtual network, such as a virtual private network ("VPN"); the Internet; an intranet; an extranet; a public switched telephone network ("PSTN"); an infra-red network; a wireless network, including, without limitation, a network operating under any of the IEEE 802.11 suite of protocols, the Bluetooth™ protocol known in the art, and/or any other wireless protocol; and/or any combination of these and/or other networks. In a particular embodiment, the network(s) 145a-145n may include an access network of the service provider (e.g., an Internet service provider ("ISP")). In another embodiment, the network(s) 145a-145n may include a core network of the service provider and/or the Internet.

[0044] In operation, at least one of computing system 105, DNS resolver 110a among the one or more DNS resolvers 110a-110n, ACL or filter 115a or 115b, threat intelligence system

125, and/or AI/ML system 130 (collectively, "computing system" or the like) may receive a first user datagram protocol ("UDP") -based DNS request (e.g., UDP request 160, or the like), the first UDP-based DNS request including a source address and a query for a destination DNS record associated with at least one of a destination device (not shown) in a network (e.g., network(s) 145, or the like), a destination entity (not shown) associated with the destination device, or a destination domain (not shown) associated with the destination device or the destination entity. In response to receiving the first UDP-based DNS request, the computing system may send a UDP-based response message (e.g., response message 165, or the like) to the source address. In some cases, the source address may include, without limitation, at least one of a source port or a source IP address, and/or the like. In some instances, the response message may include an empty payload portion and a header portion containing a truncate ("TC") flag that is set, the set TC flag in the response message may indicate for the source address to resend the first UDP-based DNS request as a transmission control protocol ("TCP") -based DNS request.

[0045] In this manner, the computing system may send the empty UDP response with the TC flag set, without performing any other DNS-based tasks until a TCP-based DNS request is received from the source address. In the case that the source address is associated with the requesting client device (e.g., source address associated with client device(s) 150b that is associated with the second entity 155b, or the like), the sending client device is more likely to be a legitimate requester and thus more likely to send a TCP-based DNS request in response to the UDP-based response with TC flag set. In the case that the source address is not associated with the requesting client device (e.g., the requesting client device(s) 150a that is associated with the first entity 155a attempts to spoof the source address of client device(s) 150b that is associated with the second entity 155b, or the like), for UDP-based responses that reach the second client device(s) 150b that is associated with the source address, the second client device(s) 150b may determine that it did not send the first UDP-based DNS request, and thus would not send a TCP-based DNS request in response. Alternatively, for UDP-based response that reach the first client device(s) 150a, in the case that first client device(s) 150a is used for sending high volume UDP-based DNS requests (e.g., as part of a distributed denial of service ("DDoS") attack, or the like), it would not likely respond to the UDP-based response as its resources are likely devoted to mass

sending of UDP-based DNS requests, and thus would not likely sent a TCP-based DNS request in response.

[0046] When a first TCP-based DNS request (e.g., TCP request 170, or the like) corresponding to the first UDP-based DNS request (e.g., UDP request 160, or the like) is received from the source address within a first predetermined period (e.g., 50, 100, 150, 200, 250, 300, 350, 400, 450, or 500 ms, or 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50, or 60 s, or 2, 3, 4, 5, 10, 15, or 20 minutes, or a period within a range between 1 and 500 ms, between 1 and 100 ms, between 1 and 50 ms, between 1 and 60 s, between 1 and 30 s, between 1 and 15 s, between 1 and 20 minutes, between 1 and 10 minutes, or between 1 and 5 minutes, or the like), the computing system may send, to the source address, a TCP-based response message comprising an answer to the query for the destination DNS record. Herein, in some examples, for the first TCP-based DNS request to correspond to the first UDP-based DNS request, at least one of a query, a query type, a query source, or a mail exchange ("MX") record of the first UDP-based DNS request should be the same as corresponding at least one of a query, a query type, a query source, or a MX record of the first TCP-based DNS request, or vice versa.

[0047] When a TCP-based DNS request corresponding to the first UDP-based DNS request is not received from the source address within the first predetermined period, the computing system may cause one or more second UDP-based DNS requests from the source address to be blocked. In some cases, the one or more second UDP-based DNS requests may include the first UDP-based DNS request.

[0048] According to some embodiments, sending the TCP-based response message may comprise: the computing system sending, to the source address, the destination DNS record, in some cases, by returning, in a DNS response (e.g., DNS response 175, or the like), the destination DNS record for the destination device being obtained from at least one of a cache (e.g., cache 120, or the like) of the computing system or an authoritative nameserver search (e.g., a search of authoritative nameserver 135 and/or corresponding database(s) 140, or the like). Alternatively, sending the TCP-based response message may comprise: the computing system sending, to the source address, a message indicating that the destination DNS record was not found.

[0049] In some embodiments, for "a grant access condition," the computing system may, after receiving the first TCP-based DNS request corresponding to the first UDP-based DNS

request, update one or more rules of at least one ACL (e.g., ACL or filter 115a or 115b, or the like) to grant access to the network to the source address (i.e., to grant the source address access to the network and thus to allow communications between the source address and the destination device, assuming the destination DNS record can be found, or the like). Alternatively, or additionally, for "a grant access condition," the computing system may, after receiving the first TCP-based DNS request corresponding to the first UDP-based DNS request, perform at least one of: causing one or more second UDP-based DNS requests that are received by the computing system from the source address to be processed; or causing a threat intelligence system (e.g., threat intelligence system 125, or the like) to communicate to each of a plurality of DNS resolvers (e.g., DNS resolvers 110b-110n among the one or more DNS resolvers 110a-110n, or the like) to allow one or more third UDP-based DNS requests that are received by the plurality of DNS resolvers from the source address to be processed. Alternatively, or additionally, for "a grant access condition," the computing system may, cause one or more fourth UDP-based DNS requests that are received, within a second predetermined period (e.g., 50, 100, 150, 200, 250, 300, 350, 400, 450, or 500 ms, or 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50, or 60 s, or 2, 3, 4, 5, 10, 15, or 20 minutes, or a period within a range between 1 and 500 ms, between 1 and 100 ms, between 1 and 50 ms, between 1 and 60 s, between 1 and 30 s, between 1 and 15 s, between 1 and 20 minutes, between 1 and 10 minutes, or between 1 and 5 minutes, or the like) after receiving the first TCP-based DNS request, from the source address to be processed. Herein, causing UDP-based DNS requests to be processed may refer to obtaining a destination DNS record(s) in response to queries in the UDP-based DNS requests from at least one of the cache of the computing system or the authoritative nameserver search, and/or to sending a message indicating that the destination DNS record(s) was(were) not found, or the like.

[0050] In some embodiments, the computing system may analyze, using a machine learning model (e.g., a ML model of AI/ML system 130, or the like), one or more fifth UDP-based DNS requests that are received after the second predetermined period, from the source address, to determine whether or not to block the one or more fifth UDP-based DNS requests. Based on such analysis, the computing system may perform one or more DNS tasks, including, but not limited to, at least one of configuring a DNS client (either by configuring static domain name resolution or by configuring dynamic domain name resolution, configuring a DNS proxy, configuring DNS spoofing, specifying a source interface for DNS packets, configuring a DNS

trusted interface, or setting a differentiated services code point ("DSCP") value for outgoing DNS packets, or the like).

[0051] In some embodiments, causing the one or more second UDP-based DNS requests from the source address to be blocked [herein also referred to as "a deny access condition"] may comprise the computing system performing at least one of: updating one or more rules of at least one ACL (e.g., ACL or filter 115a or 115b, or the like) to deny access to the network to the source address (i.e., to deny the source address from accessing the network and thus from communicating with the destination device, or the like); dropping all DNS requests from the source address; or filtering all DNS requests from the source address; and/or the like.

Alternatively, or additionally, causing the one or more second UDP-based DNS requests from the source address to be blocked may comprise the computing system causing the DNS requests that are received from the source address over a third predetermined period (e.g., 50, 100, 150, 200, 250, 300, 350, 400, 450, or 500 ms, or 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50, or 60 s, or 2, 3, 4, 5, 10, 15, 20, 30, 40, 50, 60 minutes, or 2, 3, 4, 5, 6, 12, 18, or 24 hours, or 2, 3, 4, 5, 6, or 7 days, or a period within a range between 1 and 500 ms, between 1 and 100 ms, between 1 and 50 ms, between 1 and 60 s, between 1 and 30 s, between 1 and 15 s, between 1 and 60 minutes, between 1 and 30 minutes, between 1 and 10 minutes, between 1 and 5 minutes, between 1 and 24 hours, between 1 and 18 hours, between 1 and 12 hours, between 1 and 6 hours, or between 1 and 7 days, or the like) to be blocked.

[0052] According to some embodiments, the computing system may drop any one or more sixth UDP-based DNS requests that are received from the source address after receiving the first UDP-based DNS request and before receiving the first TCP-based DNS request.

[0053] In some embodiments, the computing system may determine whether the first UDP-based DNS request is a legitimate DNS request, in some cases, by performing at least one of: determining that the first UDP-based DNS request is among a number of UDP-based DNS requests that exceeds a predetermined number of requests within a fourth predetermined period (e.g., 50, 100, 150, 200, 250, 300, 350, 400, 450, or 500 ms, or 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50, or 60 s, or 2, 3, 4, 5, 10, 15, or 20 minutes, or a period within a range between 1 and 500 ms, between 1 and 100 ms, between 1 and 50 ms, between 1 and 60 s, between 1 and 30 s, between 1 and 15 s, between 1 and 20 minutes, between 1 and 10 minutes, or between 1 and 5 minutes, or the like) and that are received from the source address; or determining that an ACL

(e.g., ACL or filter 115a or 115b, or the like) already indicates that the source address should be denied access to the network; and/or the like.

[0054] In another aspect, the computing system may receive a first UDP-based DNS request (e.g., UDP request 160, or the like), the first UDP-based DNS request including a source address and a query for a destination DNS record associated with at least one of a destination device (not shown) in a network (e.g., network(s) 145, or the like), a destination entity (not shown) associated with the destination device, or a destination domain (not shown) associated with the destination device or the destination entity. In response to receiving the first UDP-based DNS request, the computing system may send a UDP-based response message (e.g., response message 165, or the like) to the source address. In some instances, the UDP-based response message may include, but is not limited to, an empty payload portion and a header portion containing a TC flag that is set, the set TC flag in the header portion of the response message indicating for the source address to resend the first UDP-based DNS request as a TCP-based DNS request. When a TCP-based DNS request corresponding to the first UDP-based DNS request is not received from the source address within a first predetermined period (similar to the first predetermined period described above), the computing system may cause one or more second UDP-based DNS requests from the source address to be blocked. In some cases, the one or more second UDP-based DNS requests may include the first UDP-based DNS request.

[0055] When a first TCP-based DNS request (e.g., TCP request 170, or the like) corresponding to the first UDP-based DNS request (e.g., UDP request 160, or the like) is received from the source address within the first predetermined period, the computing system may allow, within a second predetermined period (similar to the second predetermined period described above) after receiving the first TCP-based DNS request, at least the first UDP-based DNS request from the source address to be processed, in some cases, by sending, to the source address, at least a TCP-based response message comprising an answer to the query for the destination DNS record. The computing system may (subsequently) receive, from the source address, one or more third UDP-based DNS requests, after the second predetermined period. The computing system may analyze, using a machine learning model (e.g., a ML model of AI/ML system 130, or the like), the one or more third UDP-based DNS requests, to determine whether or not to block the one or more third UDP-based DNS requests; and may perform one or more DNS tasks (similar to the DNS tasks described above) based on the analysis.

[0056] These and other functions of the system 100 (and its components) are described in greater detail below with respect to Figs. 2-4.

[0057] Fig. 2 is a schematic diagram illustrating a non-limiting example 200 of a grant access condition when implementing improvement to DNS security, in accordance with various embodiments.

[0058] In some embodiments, DNS resolvers 210a-210n, ACL or filter 215a, ACL or filter 215b or 215c, cache 220, threat intelligence system 225, AI/ML system 230, authoritative nameserver 235, client device(s) 250, UDP request 260, response 265, TCP request 270, and DNS response 275 of Fig. 2 may be similar, if not identical, to the one or more DNS resolvers 110 and 110a-110n, ACL or filter 115a, ACL or filter 115b, cache 120, threat intelligence system 125, AI/ML system 130, authoritative nameserver 135, client device(s) 150a or 150b, UDP request 160, response 165, TCP request 170, and DNS response 175, respectively, of system 100 of Fig. 1, and the description of these components of system 100 of Fig. 1 are similarly applicable to the corresponding components of Fig. 2.

[0059] In operation, DNS resolver 210a may receive UDP request 260 from client device(s) 250. The UDP request 260 may include, without limitation, a source address (including, but not limited to, at least one of a source port or a source IP address, and/or the like) and a query for a destination DNS record associated with at least one of a destination device (not shown) in a network (e.g., network(s) 145 of Fig. 1, or the like), a destination entity (not shown) associated with the destination device, or a destination domain (not shown) associated with the destination device or the destination entity. One of, or a combination of two or more of, DNS resolver 210a, ACL or filter 215a, router ACL or filter 215b, third party ACL or filter 215c, threat intelligence system 225, and/or AI/ML system 230 (collectively, "computing system" or the like) may send UDP-based response message 265, which is a message having an empty payload portion and having its TC flag set. The set TC flag in the UDP-based response message may indicate for the client device(s) 250 to resend the UDP request 260 as a TCP-based DNS request. In this manner, the various embodiments leverage the features of the truncate flag as a novel feature for improving DNS security. The truncate flag is conventionally used in a UDP response that exceeds 512 bytes in payload size (and thus has its payload truncated) to request or indicate resending of a UDP message (to which the UDP response is responding to) as a TCP message, so

that a corresponding TCP response (which is not bound by the 512 byte payload limitation) may be sent.

[0060] In the event that client device(s) 250 sends TCP request 270 corresponding to UDP request 260, and the computing system receives the TCP request 270 within a first window of response (e.g., the first predetermined period as described above with respect to Fig. 1, or the like) and (optionally) determines that the DNS request is legitimate, the computing system may allow UDP requests (including UDP request 260, as well as any other UDP requests received from client device(s) 250 within a second window of response (e.g., the second predetermined period as described above with respect to Fig. 1, or the like) after receiving the TCP request 270) that are received from client device(s) 250 and/or other devices associated with the source address, or the like, to be processed, thereby granting access to the network(s) (e.g., network(s) 145 of Fig. 1, or the like). For instance, the computing system may process the query for the destination DNS record, by first searching a cache (e.g., cache 220 or the like) that is communicatively coupled with (or integrated with) the computing system. If the destination DNS record is contained in the cache, the computing system may send DNS response 275b (with the destination DNS record retrieved from the cache) to the client device(s) 250. If not contained in the cache, the computing system may send a DNS request 280 relaying the query for the destination DNS record to authoritative nameserver 235. If found by the authoritative nameserver 235 (e.g., from database(s) 140 of Fig. 1, or the like), then the authoritative nameserver 235 may send DNS response 275a (with the found destination DNS record) to the computing system, which may forward DNS response 275a as DNS response 275b to the client device(s) 250. If not found, the DNS responses 275a and 275b would include a message indicating that the destination DNS record was not found.

[0061] According to some embodiments, the computing system may communicate to each of a plurality of DNS resolvers (e.g., DNS resolvers 210b-210n among the one or more DNS resolvers 210a-210n, or the like) to allow one or more other UDP-based DNS requests from the at least one of the client device(s) 250 or one or more other client devices associated with either the source address and/or the entity associated with the source address (not shown) to be processed and the query for any other destination DNS records in such other UDP-based DNS requests to be responded to by the DNS system.

[0062] These and other functions of the example 200 (and its components) are described in greater detail herein with respect to Figs. 1, 3, and 4.

[0063] Fig. 3 is a schematic diagram illustrating a non-limiting example 300 of a deny access condition when implementing improvement to DNS security, in accordance with various embodiments.

[0064] In some embodiments, DNS resolvers 310a-310n, ACL or filter 315a, ACL or filter 315b or 315c, cache 320, threat intelligence system 325, AI/ML system 330, authoritative nameserver 335, client device(s) 350, UDP request 360, and response 365 of Fig. 3 may be similar, if not identical, to the one or more DNS resolvers 110 and 110a-110n, ACL or filter 115a, ACL or filter 115b, cache 120, threat intelligence system 125, AI/ML system 130, authoritative nameserver 135, client device(s) 150a or 150b, UDP request 160, and response 165, respectively, of system 100 of Fig. 1, and the description of these components of system 100 of Fig. 1 are similarly applicable to the corresponding components of Fig. 2. Similarly, DNS resolvers 310a-310n, ACL or filter 315a-315c, cache 320, threat intelligence system 325, AI/ML system 330, authoritative nameserver 335, client device(s) 350, UDP request 360, and response 365 of Fig. 3 may be similar, if not identical, to DNS resolvers 210a-210n, ACL or filter 215a-215c, cache 220, threat intelligence system 225, AI/ML system 230, authoritative nameserver 235, client device(s) 250, UDP request 260, and response 265, respectively, of example 200 of Fig. 2, and the description of these components of example 200 of Fig. 2 are similarly applicable to the corresponding components of Fig. 3.

[0065] In operation, after sending the response message 365 (similar to sending of response message 265 in the non-limiting example of Fig. 2, or the like), in the event that client device(s) 350 fails to send a TCP request corresponding to UDP request 360 (such as TCP request 270 of Fig. 2, or the like), within the first window of response, the computing system may determine that the DNS request is not legitimate, and may block subsequent UDP requests from client device(s) 350, from devices associated with the source address, and/or from the entity with which client device(s) 350 and/or source address is associated, thereby blocking access to the network(s) (e.g., network(s) 145 of Fig. 1, or the like).

[0066] According to some embodiments, the computing system may communicate to each of a plurality of DNS resolvers (e.g., DNS resolvers 310b-310n among the one or more DNS resolvers 310a-310n, or the like) to deny access to the network(s) to one or more other UDP-

based DNS requests from the at least one of the client device(s) 350 or one or more other client devices associated with either the source address and/or the corresponding entity (not shown).

[0067] These and other functions of the example 300 (and its components) are described in greater detail herein with respect to Figs. 1, 2, and 4.

[0068] Figs. 4A-4F (collectively, "Fig. 4") are flow diagrams illustrating a method 400 for implementing improvement to DNS security, in accordance with various embodiments. Method 400 of Fig. 4A continues onto Fig. 4F following the circular marker denoted, "A," and returns to Fig. 4A following the circular marker denoted, "B." Method 400 of Fig. 4A continues onto Fig. 4C following the circular marker denoted, "C," and returns to Fig. 4A following the circular marker denoted, "D." Method 400 of Fig. 4A continues onto Fig. 4E following the circular marker denoted, "E," and returns to Fig. 4A following the circular marker denoted, "D."

[0069] While the techniques and procedures are depicted and/or described in a certain order for purposes of illustration, it should be appreciated that certain procedures may be reordered and/or omitted within the scope of various embodiments. Moreover, while the method 400 illustrated by Fig. 4 can be implemented by or with (and, in some cases, are described below with respect to) the systems, examples, or embodiments 100, 200, and 300 of Figs. 1, 2, and 3, respectively (or components thereof), such methods may also be implemented using any suitable hardware (or software) implementation. Similarly, while each of the systems, examples, or embodiments 100, 200, and 300 of Figs. 1, 2, and 3, respectively (or components thereof), can operate according to the method 400 illustrated by Fig. 4 (e.g., by executing instructions embodied on a computer readable medium), the systems, examples, or embodiments 100, 200, and 300 of Figs. 1, 2, and 3 can each also operate according to other modes of operation and/or perform other suitable procedures.

[0070] In the non-limiting embodiment of Fig. 4A, method 400, at block 405, may comprise receiving, by a computing system of a domain name system ("DNS"), a first user datagram protocol ("UDP")-based DNS request. In some cases, the first UDP-based DNS request may include, but is not limited to, a source address and a query for a destination DNS record associated with at least one of a destination device in a network, a destination entity associated with the destination device, or a destination domain associated with the destination device or the destination entity.

[0071] At block 410, method 400 may comprise, in response to receiving the first UDP-based DNS request, sending, by the computing system, a UDP-based response message to the source address, the UDP-based response message including an empty payload portion and a header portion containing a truncate ("TC") flag that is set. In some instances, the set TC flag in the UDP-based response message may indicate for the source address to resend the first UDP-based DNS request as a transmission control protocol ("TCP") -based DNS request.

[0072] In some embodiments, the computing system may include, but is not limited to, at least one of a DNS resolver, a DNS recursive resolver ("recursor"), a DNS root nameserver, a top level domain ("TLD") nameserver, an authoritative nameserver, a threat intelligence system, a threat mitigation system, a server, an artificial intelligence ("AI") system, a machine learning ("ML") system, a virtual machine ("VM"), or software running on the computing system, and/or the like. In some cases, the source address may include, without limitation, at least one of a source port or a source IP address, and/or the like.

[0073] In some examples, method 400 may continue from the process at block 410 onto the process at block 415. In other examples, method 400 may continue onto the process at block 485 in Fig. 4F following the circular marker denoted, "A," before returning to the process at block 415 in Fig. 4A, as indicated by the circular marker denoted, "B."

[0074] At block 415, method 400 may comprise, when a first TCP-based DNS request corresponding to the first UDP-based DNS request is received from the source address within a first predetermined period, sending, by the computing system and to the source address, a TCP-based response message comprising an answer to the query for the destination DNS record. Merely by way of example, in some cases, for the first TCP-based DNS request to correspond to the first UDP-based DNS request, at least one of a query, a query type, a query source, or a mail exchange ("MX") record of the first UDP-based DNS request should be the same as corresponding at least one of a query, a query type, a query source, or a MX record of the first TCP-based DNS request, or vice versa.

[0075] In some examples, method 400 may return from the process at block 415 onto the process at block 420. In other examples, method 400 may continue onto one or more of the process at block 435, the process at block 440, the process at block 445, and/or the process at block 450, or the like, as shown in Fig. 4C, each following the circular marker denoted, "C" in

Fig. 4A, before returning to the process at block 405 in Fig. 4A, as indicated by the circular marker denoted, "D."

[0076] Method 400 may further comprise, at block 420, when a TCP-based DNS request corresponding to the first UDP-based DNS request is not received from the source address within the first predetermined period, causing, by the computing system, one or more second UDP-based DNS requests from the source address to be blocked, the one or more second UDP-based DNS requests comprising the first UDP-based DNS request.

[0077] In some examples, method 400 may return from the process at block 420 onto the process at block 405. In other examples, method 400 may continue onto the process at block 475 in Fig. 4E following the circular marker denoted, "E," before returning to the process at block 405 in Fig. 4A, as indicated by the circular marker denoted, "D."

[0078] With reference to the non-limiting example of Fig. 4B, sending the TCP-based response message (at block 415) may comprise one of: sending, by the computing system and to the source address, the destination DNS record based on the first UDP-based DNS request and/or the first TCP-based DNS request, the destination DNS record being obtained from at least one of a cache of the computing system or an authoritative nameserver search (block 425); or sending, by the computing system and to the source address, a message indicating that the destination DNS record was not found (block 430).

[0079] Referring to the non-limiting example of Fig. 4C, following the circular marker denoted, "C," in Fig. 4A), method 400 may comprise, after receiving the first TCP-based DNS request corresponding to the first UDP-based DNS request, at least one of: updating, by the computing system, one or more rules of at least one access control list ("ACL") to grant access to the network to the source address (block 435); causing, by the computing system, one or more second UDP-based DNS requests that are received by the computing system from the source address to be processed (block 440); causing, by the computing system, a threat intelligence system to communicate to each of a plurality of DNS resolvers to allow one or more third UDP-based DNS requests that are received by the plurality of DNS resolvers from the source address to be processed (block 445); or causing, by the computing system, one or more fourth UDP-based DNS requests that are received, within a second predetermined period after receiving the first TCP-based DNS request, from the source address to be processed (block 450). Herein, causing UDP-based DNS requests to be processed may refer to obtaining a destination DNS

record(s) in response to queries in the UDP-based DNS requests from at least one of the cache of the computing system or the authoritative nameserver search, and/or to sending a message indicating that the destination DNS record(s) was(were) not found, or the like.

[0080] Method 400 may return to the process at block 405 in Fig. 4A following the circular marker denoted, "D."

[0081] Referring to the non-limiting example of Fig. 4D, causing the one or more second UDP-based DNS requests from the source address to be blocked (at block 420) may comprise at least one of: updating, by the computing system, one or more rules of at least one ACL to deny access to the network to the source address (block 455); dropping, by the computing system, all DNS requests from the source address (block 460); filtering, by the computing system, all DNS requests from the source address (block 465); or causing, by the computing system, all DNS requests that are received from the source address over a third predetermined period to be blocked (block 470); and/or the like.

[0082] At block 475 in Fig. 4E (following the circular marker denoted, "E," in Fig. 4A), method 400 may comprise analyzing, by the computing system and using a machine learning model, one or more fifth UDP-based DNS requests that are received after the second predetermined period, from the source address, to determine whether or not to block the one or more fifth UDP-based DNS requests; and performing, by the computing system, one or more DNS tasks based on the analysis (block 480).

[0083] Method 400 may return to the process at block 405 in Fig. 4A following the circular marker denoted, "D."

[0084] At block 485 in Fig. 4E (following the circular marker denoted, "A," in Fig. 4A), method 400 may comprise dropping, by the computing system, any one or more sixth UDP-based DNS requests that are received from the source address after receiving the first UDP-based DNS request and before receiving the first TCP-based DNS request.

[0085] Method 400 may return to the process at block 415 in Fig. 4A following the circular marker denoted, "B."

[0086] In some embodiments, method 400 may further comprise determining whether the first UDP-based DNS request is a legitimate DNS request (not shown), which may comprise at least one of: determining, by the computing system, that the first UDP-based DNS request is among a number of UDP-based DNS requests that exceeds a predetermined number of requests

within a fourth predetermined period and that are received from the source address (not shown); or determining, by the computing system, that an ACL already indicates that the source address should be denied access to the network (not shown); and/or the like.

[0087] Exemplary System and Hardware Implementation

[0088] Fig. 5 is a block diagram illustrating an exemplary computer or system hardware architecture, in accordance with various embodiments. Fig. 5 provides a schematic illustration of one embodiment of a computer system 500 of the service provider system hardware that can perform the methods provided by various other embodiments, as described herein, and/or can perform the functions of computer or hardware system (i.e., computing system 105, domain name system ("DNS") resolvers 110, 110a-110n, 210a-210n, and 310a-310n, access control list ("ACL") or filter 115a, 115b, 215a-215c, and 315a-315c, cache 120, 220, and 320, threat intelligence system 125, 225, and 325, artificial intelligence ("AI")/machine learning ("ML") system 130, 230, and 330, authoritative nameserver 135, 235, and 335, etc.), as described above. It should be noted that Fig. 5 is meant only to provide a generalized illustration of various components, of which one or more (or none) of each may be utilized as appropriate. Fig. 5, therefore, broadly illustrates how individual system elements may be implemented in a relatively separated or relatively more integrated manner.

[0089] The computer or hardware system 500 – which might represent an embodiment of the computer or hardware system (i.e., computing system 105, DNS resolvers 110, 110a-110n, 210a-210n, and 310a-310n, ACL or filter 115a, 115b, 215a-215c, and 315a-315c, cache 120, 220, and 320, threat intelligence system 125, 225, and 325, AI/ML system 130, 230, and 330, authoritative nameserver 135, 235, and 335, etc.), described above with respect to Figs. 1-4 – is shown comprising hardware elements that can be electrically coupled via a bus 505 (or may otherwise be in communication, as appropriate). The hardware elements may include one or more processors 510, including, without limitation, one or more general-purpose processors and/or one or more special-purpose processors (such as microprocessors, digital signal processing chips, graphics acceleration processors, and/or the like); one or more input devices 515, which can include, without limitation, a mouse, a keyboard, and/or the like; and one or more output devices 520, which can include, without limitation, a display device, a printer, and/or the like.

[0090] The computer or hardware system 500 may further include (and/or be in communication with) one or more storage devices 525, which can comprise, without limitation, local and/or network accessible storage, and/or can include, without limitation, a disk drive, a drive array, an optical storage device, solid-state storage device such as a random access memory ("RAM") and/or a read-only memory ("ROM"), which can be programmable, flash-updateable, and/or the like. Such storage devices may be configured to implement any appropriate data stores, including, without limitation, various file systems, database structures, and/or the like.

[0091] The computer or hardware system 500 might also include a communications subsystem 530, which can include, without limitation, a modem, a network card (wireless or wired), an infra-red communication device, a wireless communication device and/or chipset (such as a Bluetooth™ device, an 802.11 device, a Wi-Fi device, a WiMAX device, a wireless wide area network ("WWAN") device, cellular communication facilities, etc.), and/or the like. The communications subsystem 530 may permit data to be exchanged with a network (such as the network described below, to name one example), with other computer or hardware systems, and/or with any other devices described herein. In many embodiments, the computer or hardware system 500 will further comprise a working memory 535, which can include a RAM or ROM device, as described above.

[0092] The computer or hardware system 500 also may comprise software elements, shown as being currently located within the working memory 535, including an operating system 540, device drivers, executable libraries, and/or other code, such as one or more application programs 545, which may comprise computer programs provided by various embodiments (including, without limitation, hypervisors, virtual machines ("VMs"), and the like), and/or may be designed to implement methods, and/or configure systems, provided by other embodiments, as described herein. Merely by way of example, one or more procedures described with respect to the method(s) discussed above might be implemented as code and/or instructions executable by a computer (and/or a processor within a computer); in an aspect, then, such code and/or instructions can be used to configure and/or adapt a general purpose computer (or other device) to perform one or more operations in accordance with the described methods.

[0093] A set of these instructions and/or code might be encoded and/or stored on a non-transitory computer readable storage medium, such as the storage device(s) 525 described above. In some cases, the storage medium might be incorporated within a computer system, such as the

system 500. In other embodiments, the storage medium might be separate from a computer system (i.e., a removable medium, such as a compact disc, etc.), and/or provided in an installation package, such that the storage medium can be used to program, configure, and/or adapt a general purpose computer with the instructions/code stored thereon. These instructions might take the form of executable code, which is executable by the computer or hardware system 500 and/or might take the form of source and/or installable code, which, upon compilation and/or installation on the computer or hardware system 500 (e.g., using any of a variety of generally available compilers, installation programs, compression/decompression utilities, etc.) then takes the form of executable code.

[0094] It will be apparent to those skilled in the art that substantial variations may be made in accordance with specific requirements. For example, customized hardware (such as programmable logic controllers, field-programmable gate arrays, application-specific integrated circuits, and/or the like) might also be used, and/or particular elements might be implemented in hardware, software (including portable software, such as applets, etc.), or both. Further, connection to other computing devices such as network input/output devices may be employed.

[0095] As mentioned above, in one aspect, some embodiments may employ a computer or hardware system (such as the computer or hardware system 500) to perform methods in accordance with various embodiments of the invention. According to a set of embodiments, some or all of the procedures of such methods are performed by the computer or hardware system 500 in response to processor 510 executing one or more sequences of one or more instructions (which might be incorporated into the operating system 540 and/or other code, such as an application program 545) contained in the working memory 535. Such instructions may be read into the working memory 535 from another computer readable medium, such as one or more of the storage device(s) 525. Merely by way of example, execution of the sequences of instructions contained in the working memory 535 might cause the processor(s) 510 to perform one or more procedures of the methods described herein.

[0096] The terms "machine readable medium" and "computer readable medium," as used herein, refer to any medium that participates in providing data that causes a machine to operate in a specific fashion. In an embodiment implemented using the computer or hardware system 500, various computer readable media might be involved in providing instructions/code to processor(s) 510 for execution and/or might be used to store and/or carry such instructions/code

(e.g., as signals). In many implementations, a computer readable medium is a non-transitory, physical, and/or tangible storage medium. In some embodiments, a computer readable medium may take many forms, including, but not limited to, non-volatile media, volatile media, or the like. Non-volatile media includes, for example, optical and/or magnetic disks, such as the storage device(s) 525. Volatile media includes, without limitation, dynamic memory, such as the working memory 535. In some alternative embodiments, a computer readable medium may take the form of transmission media, which includes, without limitation, coaxial cables, copper wire, and fiber optics, including the wires that comprise the bus 505, as well as the various components of the communication subsystem 530 (and/or the media by which the communications subsystem 530 provides communication with other devices). In an alternative set of embodiments, transmission media can also take the form of waves (including without limitation radio, acoustic, and/or light waves, such as those generated during radio-wave and infra-red data communications).

[0097] Common forms of physical and/or tangible computer readable media include, for example, a floppy disk, a flexible disk, a hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read instructions and/or code.

[0098] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to the processor(s) 510 for execution. Merely by way of example, the instructions may initially be carried on a magnetic disk and/or optical disc of a remote computer. A remote computer might load the instructions into its dynamic memory and send the instructions as signals over a transmission medium to be received and/or executed by the computer or hardware system 500. These signals, which might be in the form of electromagnetic signals, acoustic signals, optical signals, and/or the like, are all examples of carrier waves on which instructions can be encoded, in accordance with various embodiments of the invention.

[0099] The communications subsystem 530 (and/or components thereof) generally will receive the signals, and the bus 505 then might carry the signals (and/or the data, instructions, etc. carried by the signals) to the working memory 535, from which the processor(s) 505

retrieves and executes the instructions. The instructions received by the working memory 535 may optionally be stored on a storage device 525 either before or after execution by the processor(s) 510.

[0100] While certain features and aspects have been described with respect to exemplary embodiments, one skilled in the art will recognize that numerous modifications are possible. For example, the methods and processes described herein may be implemented using hardware components, software components, and/or any combination thereof. Further, while various methods and processes described herein may be described with respect to particular structural and/or functional components for ease of description, methods provided by various embodiments are not limited to any particular structural and/or functional architecture but instead can be implemented on any suitable hardware, firmware and/or software configuration. Similarly, while certain functionality is ascribed to certain system components, unless the context dictates otherwise, this functionality can be distributed among various other system components in accordance with the several embodiments.

[0101] Moreover, while the procedures of the methods and processes described herein are described in a particular order for ease of description, unless the context dictates otherwise, various procedures may be reordered, added, and/or omitted in accordance with various embodiments. Moreover, the procedures described with respect to one method or process may be incorporated within other described methods or processes; likewise, system components described according to a particular structural architecture and/or with respect to one system may be organized in alternative structural architectures and/or incorporated within other described systems. Hence, while various embodiments are described with—or without—certain features for ease of description and to illustrate exemplary aspects of those embodiments, the various components and/or features described herein with respect to a particular embodiment can be substituted, added and/or subtracted from among other described embodiments, unless the context dictates otherwise. Consequently, although several exemplary embodiments are described above, it will be appreciated that the invention is intended to cover all modifications and equivalents within the scope of the following claims.

WHAT IS CLAIMED IS:

- 1 1. A method, comprising:
2 receiving, by a computing system of a domain name system ("DNS"), a first user
3 datagram protocol ("UDP") -based DNS request, the first UDP-based DNS request
4 comprising a source address and a query for a destination DNS record associated with
5 at least one of a destination device in a network, a destination entity associated with
6 the destination device, or a destination domain associated with the destination device
7 or the destination entity;
8 in response to receiving the first UDP-based DNS request, sending, by the computing
9 system, a UDP-based response message to the source address, the UDP-based
10 response message comprising an empty payload portion and a header portion
11 containing a truncate ("TC") flag that is set;
12 when a first transmission control protocol ("TCP") -based DNS request corresponding to
13 the first UDP-based DNS request is received from the source address within a first
14 predetermined period, sending, by the computing system and to the source address, a
15 TCP-based response message comprising an answer to the query for the destination
16 DNS record; and
17 when a TCP-based DNS request corresponding to the first UDP-based DNS request is not
18 received from the source address within the first predetermined period, causing, by
19 the computing system, UDP-based DNS requests from the source address to be
20 blocked.
- 1 2. The method of claim 1, wherein the computing system comprises at least one of a
2 DNS resolver, a DNS recursive resolver ("recursor"), a DNS root nameserver, a top level domain
3 ("TLD") nameserver, an authoritative nameserver, a threat intelligence system, a threat
4 mitigation system, a server, an artificial intelligence ("AI") system, a machine learning ("ML")
5 system, a virtual machine ("VM"), or software running on the computing system.
- 1 3. The method of claim 1, wherein the source address comprises at least one of a
2 source port or a source IP address.

- 1 4. The method of claim 1, wherein sending the TCP-based response message
2 comprises one of:
3 sending, by the computing system and to the source address, the destination DNS record,
4 the destination DNS record being obtained from at least one of a cache of the
5 computing system or an authoritative nameserver search; or
6 sending, by the computing system and to the source address, a message indicating that the
7 destination DNS record was not found.
- 1 5. The method of claim 1, further comprising:
2 after receiving the first TCP-based DNS request corresponding to the first UDP-based
3 DNS request, updating, by the computing system, one or more rules of at least one
4 access control list ("ACL") to grant access to the network to the source address.
- 1 6. The method of claim 1, further comprising, after receiving the first TCP-based
2 DNS request corresponding to the first UDP-based DNS request, performing at least one of:
3 causing, by the computing system, one or more second UDP-based DNS requests that are
4 received by the computing system from the source address to be processed; or
5 causing, by the computing system, a threat intelligence system to communicate to each of
6 a plurality of DNS resolvers to allow one or more third UDP-based DNS requests that
7 are received by the plurality of DNS resolvers from the source address to be
8 processed.
- 1 7. The method of claim 1, further comprising:
2 causing, by the computing system, one or more fourth UDP-based DNS requests that are
3 received, within a second predetermined period after receiving the first TCP-based
4 DNS request, from the source address to be processed.
- 1 8. The method of claim 7, further comprising:
2 analyzing, by the computing system and using a machine learning model, one or more
3 fifth UDP-based DNS requests that are received after the second predetermined
4 period, from the source address, to determine whether or not to block the one or more
5 fifth UDP-based DNS requests; and

6 performing, by the computing system, one or more DNS tasks based on the analysis.

1 9. The method of claim 1, wherein causing the UDP-based DNS requests from the
2 source address to be blocked comprises at least one of:

3 updating, by the computing system, one or more rules of at least one access control list

4 ("ACL") to deny access to the network to the source address;

5 dropping, by the computing system, all DNS requests from the source address; or

6 filtering, by the computing system, all DNS requests from the source address.

1 10. The method of claim 9, wherein causing the first UDP-based DNS request from
2 the source address to be blocked comprises causing, by the computing system, all DNS requests
3 that are received from the source address over a third predetermined period to be blocked.

1 11. The method of claim 1, further comprising:

2 dropping, by the computing system, any one or more sixth UDP-based DNS requests that

3 are received from the source address after receiving the first UDP-based DNS request

4 and before receiving the first TCP-based DNS request.

1 12. The method of claim 1, wherein at least one of a query, a query type, a query
2 source, or a mail exchange ("MX") record of the first UDP-based DNS request is the same as
3 corresponding at least one of a query, a query type, a query source, or a MX record of the first
4 TCP-based DNS request.

1 13. The method of claim 1, further comprising determining, by the computing system,
2 whether the first UDP-based DNS request is a legitimate DNS request, by performing at least
3 one of:

4 determining, by the computing system, that the first UDP-based DNS request is among a
5 number of UDP-based DNS requests that exceeds a predetermined number of
6 requests within a fourth predetermined period and that are received from the source
7 address; or

8 determining, by the computing system, that an ACL already indicates that the source
9 address should be denied access to the network.

1 14. A system, comprising:
2 a computing system of a domain name system ("DNS"), comprising:
3 at least one first processor; and
4 a first non-transitory computer readable medium communicatively coupled to the
5 at least one first processor, the first non-transitory computer readable medium
6 having stored thereon computer software comprising a first set of instructions
7 that, when executed by the at least one first processor, causes the computing
8 system to:
9 receive a first user datagram protocol ("UDP") -based DNS request, the
10 first UDP-based DNS request comprising a source address and a query
11 for a destination DNS record associated with at least one of a
12 destination device in a network, a destination entity associated with the
13 destination device, or a destination domain associated with the
14 destination device or the destination entity;
15 in response to receiving the first UDP-based DNS request, send a UDP-
16 based response message to the source address, the UDP-based
17 response message comprising an empty payload portion and a header
18 portion containing truncate ("TC") flag that is set;
19 when a first transmission control protocol ("TCP") -based DNS request
20 corresponding to the first UDP-based DNS request is received from
21 the source address within a first predetermined period, send, to the
22 source address, a TCP-based response message comprising an answer
23 to the query for the destination DNS record; and
24 when a TCP-based DNS request corresponding to the first UDP-based
25 DNS request is not received from the source address within the first
26 predetermined period, cause UDP-based DNS requests from the source
27 address to be blocked.

1 15. The system of claim 14, wherein the computing system comprises at least one of a
2 DNS resolver, a DNS recursive resolver ("recursor"), a DNS root nameserver, a top level domain

3 ("TLD") nameserver, an authoritative nameserver, a threat intelligence system, a threat
4 mitigation system, a server, an artificial intelligence ("AI") system, a machine learning ("ML")
5 system, a virtual machine ("VM"), or software running on the computing system.

1 16. The system of claim 14, wherein the source address comprises at least one of a
2 source port or a source IP address.

1 17. The system of claim 14, wherein causing the UDP-based DNS requests from the
2 source address to be blocked comprises at least one of:
3 updating one or more rules of at least one access control list ("ACL") to deny access to
4 the network to the source address;
5 dropping all DNS requests from the source address; or
6 filtering all DNS requests from the source address.

1 18. A method, comprising:
2 receiving, by a computing system of a domain name system ("DNS"), a first user
3 datagram protocol ("UDP") -based DNS request, the first UDP-based DNS request
4 comprising a source address and a query for a destination DNS record associated with
5 at least one of a destination device in a network, a destination entity associated with
6 the destination device, or a destination domain associated with the destination device
7 or the destination entity;
8 in response to receiving the first UDP-based DNS request, sending, by the computing
9 system, a UDP-based response message to the source address, the UDP-based
10 response message comprising an empty payload portion and a header portion
11 containing a truncate ("TC") flag that is set;
12 when a first transmission control protocol ("TCP") -based DNS request corresponding to
13 the first UDP-based DNS request is not received from the source address within a
14 first predetermined period, causing, by the computing system, one or more second
15 UDP-based DNS requests from the source address to be blocked, the one or more
16 second UDP-based DNS requests comprising the first UDP-based DNS request;
17 when a first TCP-based DNS request corresponding to the first UDP-based DNS request
18 is received from the source address within the first predetermined period, allowing,

19 by the computing system and within a second predetermined period after receiving
20 the first TCP-based DNS request, at least the first UDP-based DNS request from the
21 source address to be processed, by sending, to the source address, at least a TCP-
22 based response message comprising an answer to the query for the destination DNS
23 record;
24 receiving, by the computing system and from the source address, one or more third UDP-
25 based DNS requests, after the second predetermined period;
26 analyzing, by the computing system and using a machine learning model, the one or more
27 third UDP-based DNS requests, to determine whether or not to block the one or more
28 third UDP-based DNS requests; and
29 performing, by the computing system, one or more DNS tasks based on the analysis.

1 19. The method of claim 18, wherein the computing system comprises at least one of
2 a DNS resolver, a DNS recursive resolver ("recursor"), a DNS root nameserver, a top level
3 domain ("TLD") nameserver, an authoritative nameserver, a threat intelligence system, a threat
4 mitigation system, a server, an artificial intelligence ("AI") system, a machine learning ("ML")
5 system, a virtual machine ("VM"), or software running on the computing system.

1 20. The method of claim 18, wherein causing the one or more second UDP-based
2 DNS requests from the source address to be blocked comprises at least one of:
3 updating, by the computing system, one or more rules of at least one access control list
4 ("ACL") to deny access to the network to the source address;
5 dropping, by the computing system, all DNS requests from the source address; or
6 filtering, by the computing system, all DNS requests from the source address.