United States Patent Application Publication	20250267122
Kind Code	A1
Publication Date	August 21, 2025
Inventor(s)	Szurdi; Janos et al.

DETECTING SHADOWED DOMAINS

Abstract

A method and system for detecting shadowed domains is provided. New hostnames are collected for a predetermined period of time. Candidate shadowed domains are selected from the new hostnames. Classification of the candidate shadowed domains is performed based on a plurality of features relating to the candidate shadowed domains to output a set of identified shadowed domains. An action is performed based on the set of identified shadowed domains.

Inventors: **Szurdi; Janos (Sunnyvale, CA), Houser; Rebekah (Sunnyvale, CA), Liu; Daiping (Santa Clara, CA)**

Applicant: **Palo Alto Networks, Inc.** (Santa Clara, CA)

Family ID: **1000008586608**

Appl. No.: **19/184917**

Filed: **April 21, 2025**

Related U.S. Application Data

parent US continuation 17878665 20220801 parent-grant-document US 12309116 child US 19184917

Publication Classification

Int. Cl.: **H04L9/40** (20220101); **H04L41/16** (20220101)

U.S. Cl.:

CPC   **H04L63/0236** (20130101); **H04L41/16** (20130101); **H04L63/0263** (20130101);

Background/Summary

BACKGROUND OF THE INVENTION
[0002] Malicious software (malware) generally refers to unwanted, hostile, or intrusive software that can be used to disrupt computer or network operations, collect private or sensitive information, or access private computer systems or networks. Malware can be in the form of executable code, scripts, active content, and other software. Example malware includes computer viruses, worms, Trojan horses, rootkits, keyloggers, spyware, adware, botnet command and control (C&C) related malware, and other unwanted, hostile, or intrusive software.
[0003] Security solutions (e.g., security devices or appliances, which can provide http://en.wikipedia.org/wiki/Firewall_(computing) firewall solutions) can be used to safeguard against malware. For example, a firewall can identify and prevent the further spread of malware in a network.
[0004] A firewall generally protects networks from unauthorized access while permitting authorized communications to pass through the firewall. A firewall is typically implemented as a device or a set of devices, or software executed on a device, such as a computer or appliance, that provides a firewall function for network access. For example, firewalls can be integrated into operating systems of devices (e.g., computers, smart phones, tablets, or other types of network communication capable devices). Firewalls can also be integrated into or executed as software on servers, gateways, network/routing devices (e.g., network routers), or appliances (e.g., security appliances or other types of special purpose devices).
[0005] Firewalls typically deny or permit network transmission based on a set of rules. These sets of rules are often referred to as policies. For example, a firewall can filter inbound traffic by applying a set of rules or policies. A firewall can also filter outbound traffic by applying a set of rules or policies. Firewalls can also be capable of performing basic routing functions.

Description

BRIEF DESCRIPTION OF THE DRAWINGS
[0006] Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.
[0007] FIG. **1** is a functional block diagram illustrating an architecture for detecting shadowed domains in accordance with some embodiments.
[0008] FIG. **2** is an example of a list of shadowed domains and benign domains.
[0009] FIG. **3** is an example of a unique resource record (RR) entry.
[0010] FIG. **4** is an example of a workflow for detecting shadowed domains.
[0011] FIG. **5**A is an example of a workflow for performing trained machine learning shadowed domain classification.
[0012] FIG. **5**B is an example of a neural network.
[0013] FIG. **6**A is a flow diagram illustrating an embodiment of a process for detecting shadowed domains.
[0014] FIG. **6**B is a flow diagram illustrating an embodiment of a process for collecting new hostnames for a predetermined period of time.
[0015] FIG. **7** is a flow diagram illustrating an embodiment of a process for performing post-processing on a set of identified shadowed domains.

[0016] FIG. **8** is a flow diagram illustrating another embodiment of a process for performing post-processing on a set of identified shadowed domains.

[0017] FIG. **9** is a flow diagram illustrating yet another embodiment of a process for performing post-processing on a set of identified shadowed domains.

DETAILED DESCRIPTION

[0018] The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0019] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

[0020] Malicious software (malware) generally refers to unwanted, hostile, or intrusive software that can be used to disrupt computer or network operations, collect private or sensitive information, or access private computer systems or networks. Malware can be in the form of executable code, scripts, active content, and other software. Example malware includes computer viruses, worms, Trojan horses, rootkits, keyloggers, spyware, adware, botnet command and control (C&C) related malware, and other unwanted, hostile, or intrusive software.

[0021] Security solutions (e.g., security devices or appliances, which can provide firewall solutions) can be used to safeguard against malware. For example, a firewall can identify and prevent the further spread of malware in a network.

[0022] A firewall generally protects networks from unauthorized access while permitting authorized communications to pass through the firewall. A firewall is typically implemented as a device or a set of devices, or software executed on a device, such as a computer or appliance, that provides a firewall function for network access. For example, firewalls can be integrated into operating systems of devices (e.g., computers, smart phones, tablets, or other types of network communication capable devices). Firewalls can also be integrated into or executed as software on servers, gateways, network/routing devices (e.g., network routers), or appliances (e.g., security appliances or other types of special purpose devices).

[0023] Firewalls typically deny or permit network transmission based on a set of rules. These sets of rules are often referred to as policies. For example, a firewall can filter inbound traffic by applying a set of rules or policies. A firewall can also filter outbound traffic by applying a set of rules or policies. Firewalls can also be capable of performing basic routing functions.

[0024] A network domain generally refers to a domain that is identified by a domain name. A domain name typically includes an identification string (e.g., www.example-web-site.com) that defines a realm of authority or control for a domain on the Internet. Domain names are generally

formed by rules and procedures of the Domain Name System (DNS). A domain name can be registered in the DNS as a domain name.

[0025] Network domains can be used in various networking contexts and application-specific naming and addressing purposes. For example, a domain name can be used to identify an Internet Protocol (IP) resource, such as a web site (e.g., a server hosting a web site), or any other service accessible via the Internet (e.g., a File Transfer Protocol (FTP) resource or other services accessible via the Internet).

[0026] A DNS service can be used to translate a domain name into an IP address. For example, when a user types in a domain name (e.g., an Internet or Intranet domain name), such as example.com, using a web browser, an authoritative DNS server can translate the domain name into an IP address, such as 172.16.254.1 (for IPv4) and 2001: db8: 0:1234:0:567:8:1 (for IPv6). However, if a user attempts to perform a DNS lookup or host command on an unregistered or invalid domain name, such as examplefakedomain.com, then an error can be received indicating that such is a non-existing domain name, or in other words, that such is a non-existent domain (NXDOMAIN). Generally, an NXDOMAIN (e.g., an NXDOMAIN response received in response to a DNS query for a given domain name) is a condition or error that can be indicated for an Internet domain name that is unable to be resolved using the DNS servers (e.g., invalid domain name) or that the Internet domain name is not yet registered. In some cases, an NXDOMAIN can also be indicated due to a network or DNS server problem.

[0027] Network domains can also be used by malware. For example, malware can be distributed or propagated using a network domain, such as www.bad-malware-download-site.com. As another example, botnet C&C related malware can be associated with a network domain, such as www.botnet-site.com. Various commercial efforts as well as open project efforts exist to provide listings of network domains (e.g., bad/malware domains) that are known to be used to distribute or propagate malware. Some approaches use such network domain listings (e.g., bad/malware domain listings) to provide spoofed replies, in response to any requests to a network domain on such a listing, as a mechanism for preventing propagation of malware distribution.

[0028] Blackholing is a technique that can be used to send all traffic (e.g., network communications) to a network domain (e.g., DNS or IP address) to a null interface or non-existent server (e.g., sometimes referred to as a black hole). For example, an Internet Service Provider (ISP) can manage such blackholing for efficiency and to avoid affecting network connectivity. However, while blackholing may be efficient for certain types of severe network attacks, such an approach fails to allow for analysis of the traffic to the blackholed network domain.

[0029] Sinkholing generally refers to a technique for routing traffic (e.g., network communications) to a valid IP address. For example, a security device (e.g., a network device, which can be implemented using a server or appliance) associated with the valid IP address can receive the traffic that was directed (e.g., redirected) to the sinkholed IP address that is associated with the network domain. The security device that receives the traffic can analyze the traffic. Based on the analysis of the traffic, the security device can then perform an action (e.g., reject bad packets or perform some other action).

[0030] For example, malware, such as C&C malware and/or other types of malware, can use domain names for various nefarious purposes including communication with command and control servers and phishing. To perpetrate these activities, cybercriminals can either purchase domain names (malicious registration) or compromise existing ones (DNS hijacking/compromise).

[0031] Domain shadowing is a subcategory of DNS hijacking where cybercriminals attempt to stay unnoticed after hijacking a domain name. First, cybercriminals stealthily insert subdomains—called shadowed domains—under a compromised domain name. Second, the cybercriminals keep existing records to allow normal operation of services such as websites, email servers, and any other services that were using a compromised domain. By ensuring the undisturbed operation of existing services, the cybercriminals can make the compromised domain inconspicuous to domain owners

and a cleanup of malicious entries unlikely. As a result, domain shadowing provides cybercriminals access to virtually unlimited subdomains inheriting the compromised domain's benign reputation.

[0032] What is needed are new and improved techniques for detecting shadowed domains.

[0033] Accordingly, techniques for shadowed domain detection are provided.

[0034] In some embodiments, a system/method/computer program product for detecting shadowed domains includes collecting new hostnames for a predetermined period of time; selecting candidate shadowed domains from the new hostnames; performing classification of the candidate shadowed domains based on a plurality of features relating to the candidate shadowed domains to output a set of identified shadowed domains; and performing an action based on the set of identified shadowed domains.

[0035] In some embodiments, the collecting of the new hostnames for the predetermined period of time includes determining whether a newly observed hostname (NOH) from a new hostnames dataset is found in an allowlist; and in the event that the NOH is found in the allowlist, determining that the NOH is not shadowed.

[0036] In some embodiments, the performing of the classification of the candidate shadowed domains includes extracting the plurality of features relating to the candidate shadowed domains; and performing the classification of the candidate shadowed domains using a model or a set of rules.

[0037] In some embodiments, the model is a machine learning model.

[0038] In some embodiments, the model is a machine learning model, the machine learning model being a neural network.

[0039] In some embodiments, the system/method/computer program product further includes before the performing of the action, performing the post-processing on the set of identified shadowed domains, including: comparing a subnetwork of an IP address of an identified shadowed domain with a subnetwork of an IP address of a root domain associated with the identified shadowed domain; and in response to a determination that the subnetwork of an IP address of the identified shadowed domain matches the subnetwork of an IP address of the root domain associated with the identified shadowed domain, determining that the identified shadowed domain is likely benign.

[0040] In some embodiments, the subnetwork corresponds to the first 24 bits of the IP address of the identified shadowed domain.

[0041] In some embodiments, the comparing of the subnetwork of an IP address of the identified shadowed domain with the subnetwork of an IP address of the root domain associated with the identified shadowed domain includes comparing the subnetwork of an IP address of the identified shadowed domain with the subnetwork of an IP address of the root domain associated with the identified shadowed domain based on an active DNS dataset and a passive DNS dataset.

[0042] In some embodiments, the system/method/computer program product further includes before the performing of the action, performing post-processing on the set of identified shadowed domains, including: comparing a time since creation of an identified shadowed domain with a predefined threshold; and in response to a determination that the time since creation of the identified shadowed domain is equal to or exceeds the predefined threshold, determining that the identified shadowed domain is not shadowed.

[0043] In some embodiments, the system/method/computer program product further includes before the performing of the action, performing post-processing on the set of identified shadowed domains, including: comparing a time since the root domain of an identified shadowed domain was registered with a predefined threshold; and in response to a determination that the time since the root domain of the identified shadowed domain was registered is less than or equal to the predefined threshold, determining that the identified shadowed domain is not shadowed.

[0044] FIG. **1** is a functional block diagram illustrating an architecture for detecting shadowed domains in accordance with some embodiments. In some embodiments, the architecture **100**

includes a customer **110**, a Firewall **120**, The Internet **130**, a DNS Security Module **140** including a DNS Telemetry Module **145**, a passive DNS (pDNS) Module **150**, a Whois Module **160**, a Shadowed Domain Batch Detection Pipeline **170**, and an Advanced URL Filtering Module **180**.

[0045] In operation **1**, the customer **110** sends a DNS request to the Firewall **120**.

[0046] In operation **2**, the Firewall **120** forwards the DNS request to the Internet **130** so that the DNS request can be resolved, and also forwards the DNS request to the DNS Security Module **140**.

[0047] In some embodiments, the DNS Security Module **140** forwards batch DNS data to the pDNS Module **150** et al. The pDNS Module **150**, the Whois Module **160**, et al. can send the data to the Shadowed Domain Batch Detection Pipeline **170** which provides shadowed domain batch information to the DNS Security Module **140** and the Advanced URL Filtering Module **180**.

[0048] In some embodiments, the Advanced URL Filtering Module **180** sends its output to the Firewall **120**.

[0049] In operation **3**, the DNS Security Module **140** sends its output (malicious or benign) to the Firewall **120**.

[0050] In operation **4**, the Firewall **120** receives a DNS response from the Internet **130**.

[0051] In operation **5**, based on the DNS response and the outputs of the DNS Security Module **140** and the Advanced URL Filtering Module **180**, the Firewall **120** sends a DNS response to the customer **110** and the DNS Security Module **140**. In some embodiments, the DNS Telemetry Module **145** collects DNS responses, stores the DNS responses, and forwards the DNS responses to various pDNS related modules such as, for example, a New Hostnames module, a pDNS module, a Unique RR module, etc.

[0052] After the outputs of the DNS Security Module **140** and the Advanced URL Filtering Module **180**, the Firewall **120** can block shadowed domains.

[0053] FIG. **2** is an example of a list of shadowed domains and benign domains.

[0054] In the list **200**, domains spjcww.angell.vantagetennis.co.uk, tey9s9.angell.vantagetennis.co.uk, to5rsd.angell.vantagetennis.co.uk, n5pb9t.angell.vantagetennis.co.uk, and 5c6kut.angell.vantagetennis.co.uk are malicious domains. The IP address of these malicious domains have the same IP country code (CC) and were created around the same time. The other domains in the list **200** represent benign domains.

[0055] As an example, the root domain vantagetennis.co.uk was registered in 2013 and a few benign subdomains were created under the root domain vantagetennis.co.uk (e.g., www.vantagetennis.co.uk and benign-example1.vantagetennis.co.uk). IP addresses of the benign domains are all located in the same/24 subnet. As an aspect, malicious subdomains (spjcww.angell.vantagetennis.co.uk, tey9s9.angell.vantagetennis.co.uk, to5rsd.angell.vantagetennis.co.uk, n5pb9t.angell.vantagetennis.co.uk, and 5c6kut.angell.vantagetennis.co.uk) follow a homogeneous naming pattern and use IP addresses located in a different country (FR) from the country (GB) of the IP address of the root domain. In addition, the malicious subdomains were created around the same time.

[0056] FIG. **3** is an example of a unique resource record (RR) entry. The unique RR entry **300** is an entry of a passive DNS (pDNS) dataset. Entries within the pDNS dataset are collected from external sources (e.g. a third-party provider of pDNS data) and internal sources (e.g. network security devices), and deduplicated so that the pDNS dataset includes one entry per RR. Passive DNS replication typically ignores the class and time to live (TTL) of an RR, so a unique entry in this dataset is defined by rrname (owner), rrtype (type), and rrdata (data) of the RR (corresponding to rrname, rrtype and rrdata fields in the pDNS entry). Over 90 RR types exist. Some of the commonly used RR types are A, AAAA, CNAME, NS and MX. The data of an A resource record relates to the IP address. The Unique RR entry can also include metadata such as first time seen and last time seen by a source, and the number of times an RR was intercepted by a single source (e.g. a firewall, DNS resolver or other equipment configured to collect pDNS data) or by all sources combined (total count).

[0057] FIG. **4** is an example of a workflow for detecting shadowed domains. In some embodiments, the workflow **400** is implemented by the shadowed domain detection pipeline **170** of FIG. **1** and includes six stages.

[0058] In the first stage, the detection pipeline generates an allowlist (or more than one allowlist) that includes domains unlikely to be shadowed and also includes domains where it would be costly to calculate their features because they typically have a large number of subdomains.

[0059] During the first stage, the allowlist aids the detection pipeline in removing domains from the analysis that are likely benign and/or for which calculating features would be computationally expensive. Four examples of allowlists are provided.

[0060] In a first allowlist example, the detection pipeline collects popular root domains from historical records in the unique RR dataset, where popular relates to having a large number of subdomains (e.g., 100,000, or more subdomains) under them. By filtering a small fraction of the popular root domains, in some cases, a majority of subdomains observed do not need to be processed. These popular root domains are less likely to be shadowed because their owner's operational security is typically better than an owner's operational security of less popular root domains.

[0061] In a second allowlist example, some popular root domains have a large number of subdomains added to them in a recent time frame (e.g., one week). Filtering a few of these root domains having a large number of recently added subdomains can result in a large reduction in the number of subdomains that the detection pipeline is to process. These recently popular root domains can be collected from the new hostnames dataset. In some embodiments, the new hostnames dataset is derived from pDNS data sources. In some embodiments, when the sources for the pDNS dataset report an RR with a rrname (hostname) that does not appear in the pDNS data sources already, information about that rrname is added to the new hostnames dataset.

[0062] In a third allowlist example, complementing the first and second allowlist examples, the detection pipeline can also filter known dynamic DNS root domains where users can create subdomain names at will. These dynamic DNS root domains cannot be shadowed by definition.

[0063] In a fourth allowlist example, the detection pipeline looks for popular subdomains where a popular subdomain indicates that many root domains have the same subdomain created under them. For example, popular subdomains include www, smtp, ns1, and ns2. By filtering out these popular subdomains, the detection pipeline can reduce the potential for false positives later on. In some embodiments, popular subdomains are collected over a longer period of time (e.g., weeks) from the new hostnames dataset.

[0064] As an example, the allowlisting helps filter the fully qualified domain name (FQDN) dakdkadlk321.dyndns.com of FIG. **2** because the domain is a dynamic DNS provider. In another example, the allowlisting helps filter popular subdomains such as, for example, www.index.hu and www.vantagetennis.co.uk.

[0065] In a second stage, the detection pipeline collects newly observed hostnames (NOHs) that are not in the allowlists from the new hostnames dataset. In some embodiments, the detection pipeline selects newest entries in a predetermined period of time (e.g., one day) from the new hostnames dataset. In some embodiments, the detection pipeline selects domain names that are not present in any of the allowlists.

[0066] In some embodiments, the detection pipeline retains only those new hostnames that are likely to be shadowed. A new hostname can be a candidate shadowed domain if the new hostname does not have any IP address in the same/24 subnets as its root domain's IP addresses. In addition, the detection pipeline filters domain names that only resolve to non-routable IP addresses, for example, private IP addresses (e.g. 10.X.X.X, 192.168.X.X) or erroneous IP addresses (e.g., 2.4.8.256). As an example, a candidate selection benign- example1.vantagetennis.co.uk would not be retained because the candidate has an IP address in the same/24 subnet as the root domain vantagetennis.co.uk.

[0067] In a third stage, the detection pipeline selects candidate shadowed domains from the NOHs using a unique RR dataset. As an example, a candidate shadowed domain does not have an IP address in the same subnet as its root domain's IP address(es).

[0068] In a fourth stage, the detection pipeline extracts features relating to the candidate shadowed domains.

[0069] For all candidate shadowed domains found in the third stage, the detection pipeline collects and calculates features to be used as inputs for a machine learning model. The features can be divided into three groups of features. The first group of features can relate to the candidate shadowed domain itself. The second group of features can describe a root domain of the candidate shadowed domain. The third group of features can correspond to IP addresses of the candidate shadowed domain. The three groups can be called, respectively: FQDN-level features, apex aggregation features, and IP aggregation features.

[0070] In some embodiments, the features are normalized to work with different types of machine learning algorithms that can be used. In some embodiments, log values of some features—that can have very large values—are used to scale the values before normalization. In some embodiments, for the calculation of some of the features, sampling is performed to decrease the cost of computation. As an example, some IP addresses are used by tens or hundreds of millions of domain names. To approximate IP aggregate features for these IP addresses, a subset of the FQDNs (e.g., 10,000) are randomly sampled.

[0071] As an example, feature values for spjcww.angell.vantagetennis.co.uk before normalization are obtained.

1. FQDN-Level Features

[0072] Is popular subdomain: is the domain a popular subdomain? If the domain is a popular subdomain, then the value is one, otherwise, the value is zero. Note: not all popular subs are filtered. For example, the hundred most popular subdomains can be filtered using a set of allowlists and the two hundred most popular subdomains can be considered as popular for this feature. This example is not a popular subdomain so the feature is zero. [0073] IP/CC/ASN count: the number of IP addresses the candidate domain resolves to and the number of ASNs/CCs these IPs reside in. The example has one IP address so all three counts are ones. [0074] IP/CC/ASN deviation: how different are IP addresses of the FQDN as compared to the root domain. In the case of IP address deviation, the longest common prefix between the root's common prefix and a prefix of a candidate's IP address (in binary) is used and the longest common prefix is subtracted from 32 to obtain the maximum possible value. CC and ASN deviations are zero if two IP addresses are in the same ASN/CC, one if the CC and ASN deviations are in different ASNs/CCs. For all three deviation features, the minimum deviations are calculated among all possible pairings between the candidate's and the root domain's IP addresses. The example has a vastly different IP address compared to the root domain's IP address, therefore, this feature will be 32. The ASN and CC deviation both will be one. However the IP deviation will be one between 94.136.40.103 and 94.136.40.102 as they are nearly the same. [0075] 94.136.40.103:01011110.10001000.00101000.01100111 [0076] 178.32.131.185:10110010.00100000.10000011.10111001 [0077] 94.136.40.102:01011110.10001000.00101000.01100110. [0078] Number of days between the first day the candidate domain was seen and the first time the root was seen in pDNS. For this example, the number of days is 532. [0079] Number of words in the subdomain and FQDN. These features are zero and three respectively. [0080] Entropy of a subdomain and average entropy of strings in FQDN. In this example, the entropy of the subdomain is high because it is randomly generated but, for the entire FQDN, the entropy is not very high because it includes many words. [0081] Prefix length. 12 is the prefix length for this example. [0082] Prefix level count. The prefix level count is two in this example. [0083] Is Apex wildcarding. This feature is an example indicating that the root domain is wildcarding.

Statistics: Example "statistics: " the minimum, the maximum, the average, the median, a unique count and/or Jeffrey divergence can be calculated.

2. Apex Aggregated Features

[0084] Number of subdomains: the number of subdomains created under the root domain. Seven in this example. [0085] Number of popular subdomains. One in this example. [0086] Ratio of popular subdomains to all subdomains. One over seven in this example. [0087] Average IP address count. One in this example, as all subdomains have only one IP address. [0088] Average total traffic. 98 in this example. [0089] Minimum/Average number of days between the first day a subdomain was seen and the first time a root domain was seen in pDNS. Around 2 years in days. [0090] CC deviation count and ratio of total CC deviation domains to subdomain count. In the example, there are two benign and five shadowed domains. The deviation count is five and the ratio is five over seven. [0091] ASN deviation count and ratio of total ASN deviation to subdomain count. In this example, ASN deviation count and ratio correspond with the CC deviation count and ratio. [0092] IP deviation statistics. The two benign domains have an IP deviation around zero while the malicious domains have an IP deviation of 32. The statistics will be calculated for the vector [0,1,32,32,32,32,32]. [0093] Prefix length statistics. In the example, statistics will be calculated for [3,15,12,12,12,12,12]. [0094] Prefix level count statistics. In the example, the prefix level count statistics is: [1,1,2,2,2,2,2] [0095] Sub domain entropy statistics. In the example, a vector of two small numbers and five large numbers is to be used. [0096] IP count statistics. In this example, it will be all ones. [0097] Total DNS resolutions count statistics. The number of times requests to the domains were observed. [0098] Total active day statistics. Active days is the number of days between the first seen day and last seen day. [0099] Resolution count per active days statistics. [0100] Resolution count per IP count statistics. [0101] IP count per active day statistics. [0102] First and last seen unique count and Jeffrey divergence.

[0103] IP aggregated feature examples are to be for IP Address 178.32.131.185. Please note that the feature examples are simplified, and in the real world, domains are often mapped to multiple IP addresses and vice-versa.

3. IP Aggregated Features

[0104] FQDN count. In this example, the FQDN count is five. [0105] Apex count. In this example, the apex count is one. [0106] Apex count per FQDN count. One over five is the Apex count per FQDN count in the example. [0107] Number of popular subdomains. Zero popular subdomains in the example. [0108] Ratio of popular subdomains to all subdomains. Zero in the example. [0109] FQDN under wildcarding roots count. Five in the example. [0110] FQDN under wildcarding roots ratio. One in the example (5/5). [0111] Wildcarding root count. One wildcarding root in the example. [0112] Wildcarding root ratio. One wildcarding root ratio in the example. [0113] Average IP count. One in the example. [0114] Average total resolution count. Because most domains in the example have a resolution count below a hundred, thus this feature would also be less than one hundred. [0115] Total DNS resolution count statistics: As a domain can have multiple IP addresses, we can sum or average the number of resolutions. [0116] IP count statistics. Example: the IP count statistics are [1,1,1,1,1]. [0117] CC deviation count and ratio: five and one (5/5) in the example. [0118] ASN deviation count and ratio: five and one (5/5) in the example. [0119] IP deviation statistics: Example [32,32,32,32,32]. [0120] Prefix length statistics. Example [12,12,12,12,12]. [0121] Prefix level count statistics. Example [2,2,2,2,2]. [0122] IP count statistics. Example [1,1,1,1,1]. [0123] Total active day statistics. [0124] Resolution count per active days statistics. [0125] Resolution count per IP count statistics. [0126] IP count per active day statistics. [0127] First and last seen unique count and Jeffrey divergence.

[0128] Since there can be multiple IP addresses for each candidate shadowed domain, the minimum, the average, and the maximum of each IP aggregate feature listed above are calculated over all the IP addresses a candidate domain is mapped to. Altogether more than three hundred features can be calculated.

[0129] For spjcww.angell.vantagetennis.co.uk, the following features can be indicative of domain shadowing: [0130] IP deviation is high. [0131] ASN and CC deviation is one. [0132] Days to root feature is high. [0133] Most FQDNs in Apex and IP aggregation have a high value for the previously mentioned four features so the corresponding aggregate features will be high too. [0134] Ratio of popular subdomains is low. [0135] Root is wildcarding. [0136] Distribution of IP count, prefix levels, and other statistics are concentrated on a few values.

[0137] In a fifth stage, the detection pipeline classifies the candidate shadowed domains as shadowed domains or not using the extracted features. In some embodiments, a trained machine learning model is used to perform the classification.

[0138] In a sixth stage, before returning a malicious or benign verdict for each candidate shadowed domain, the detection pipeline post-processes the candidate shadowed domains to obtain identified shadowed domains to reduce the likelihood of false positives.

[0139] Because the classifier in the fifth stage does not have perfect accuracy and data provided during production can have a significantly different distribution compared to labeled training data, post-processing is to be performed before a verdict is returned.

[0140] To post-process domains classified as shadowed, active DNS, Whois, and pDNS records can be collected. In some embodiments, active DNS is used to complement data missing from pDNS. For example, if an IP address of a root domain is not seen in pDNS (the IP address of the root domain is the same as an IP address of a candidate shadowed domain), then a domain can be falsely classified as shadowed. Using active DNS, the false classification can be corrected.

[0141] In some embodiments, Whois records are checked to determine registration dates of domain names because candidate domain names under root domains registered recently are more likely to be malicious rather than to be shadowed.

[0142] In some embodiments, pDNS data is used to complement Whois records in the event that the registration dates of the domain names are not available. Additionally, a subdomain or its root domain is checked to determine whether the subdomain or its root domain is very popular in terms of DNS traffic, because a false positive is more likely for these popular domain names.

[0143] As an example, the post-processing of stage six would remove benign-example2.index.hu because active DNS can find a record that is in the same/24 as a root domain and then the domain name benign-example2.index.hu would be filtered.

[0144] In some embodiments, the workflow **400** is implemented in a big data setting to allow scalable processing of billions of records and terabytes of data. For example, the workflow **400** is implemented using Google's Big Query combined with processing on virtual machines in the cloud.

[0145] FIG. **5**A is an example of a workflow for performing trained machine learning shadowed domain classification. In some embodiments, the workflow **500** is a high-level overview of a machine learning classification pipeline used to perform a classification, as in stage **5** of FIG. **4**. In some embodiments, the machine learning classification pipeline includes a trained machine learning model **510** and a class thresholds module **520**.

[0146] In some embodiments, feature selection is performed to forward only the most useful features that are input into the trained machine learning model **510** via a feature vector to increase its performance.

[0147] In some embodiments, a Chi-squared test is used for selecting features individually for the machine learning model **510**. As many of the features are highly correlated with each other, a mutual correlation (e.g., Pearson) is calculated between all features. Initially all features are considered as candidate features. Using the Chi-squared test, the best feature is selected and the rating of the other features, based on how correlated they are with the selected best feature, is decreased. The best feature is added to a list of selected features and removed from the candidate feature list. Features are iteratively selected from the candidate list until the desired number of features is obtained. The more correlated a feature is to a previously selected feature, the less useful

the feature would be to select as an additional feature to use.

[0148] In some embodiments, a Principal Component Analysis (PCA) technique is used to reduce the number of features. The PCA technique is a dimensionality reduction algorithm that results in a small number of very useful features that explain most of the variation in the data. However, the PCA technique creates new features that are harder to interpret than the features previously generated.

[0149] In some embodiments, a feature vector is input into the machine learning model **510**. In some embodiments, the values of the various features of the feature vector are normalized, so that, for example, each value is between 0 and 1. In some embodiments, the machine learning model **510** is trained using previously known shadowed domains and benign domain names. In some embodiments, the trained machine learning model **510** outputs a probability of how likely a candidate domain is shadowed. The machine learning model **510** can be any machine learning classifier such as XGboost, a neural network, or a Random Forest. After being trained, the trained machine learning model **510** is input with the feature vector and outputs a vector that corresponds to class probabilities (e.g., [0.6,0.4] for shadowed and not shadowed respectively).

[0150] In some embodiments, the class thresholds module **520** receives the vector that corresponds to class probabilities and outputs a final classification: shadowed or not shadowed.

[0151] A threshold (e.g., 0.5) is used to provide the desired false positive and false negative tradeoff. Typically, the threshold is set to make only a few false positives. Afterwards, the final classification can be output.

[0152] In some embodiments, the machine learning model **510** corresponds to a random forest classifier including a limited number of trees where each tree has a limited number of features and a limited depth. By limiting the depth of trees and the number of features, the constraints can avoid overfitting and increase the ability to generalize the machine learning model **510**. Based on the use case, machine learning models with various performance and complexity tradeoffs can be used. Some machine learning models can achieve 99.99 accuracy, 99.92 Precision and 99.87 recall using as few as 64 features and allowing each of 200 trees in the random forest to use no more than eight features and to have a maximum depth of four. A simpler machine learning model—using only 32 features where each tree can only use at most four features and have a depth of two—can also achieve 99.78 accuracy with 99.87 precision and 92.58 recall. Other machine learning model settings can be set where the constraints provide different tradeoffs to satisfy other use cases.

[0153] FIG. **5**B is an example of a neural network. In some embodiments, the neural network **550** can correspond with the machine learning model **510** of FIG. **5**A. Vector x (e.g., [0, . . . 0.32, 0.01, 1.00] representing a feature vector) can be an input layer for the neural network **550**. In some embodiments, values of each element of the feature vector are normalized, so that the value is between 0 and 1. The input layer is fed into the hidden layer. The output of the hidden layer is computed as a=σ(W.Math.x+b). W can be a weight matrix of the hidden layer where each row represents a neuron and each value in a row can represent an importance of an input value for the neuron. As an example, if the hidden layer has 100 neurons and the input layer includes 32 numeric inputs, then the weight matrix is 100×32 matrix. The function σ can be any nonlinear function, such as, for example, sigmoid, tan h, and ReLu functions. The nonlinear functions allow the neural network **550** to learn nonlinear functions between the input and output. The output layer can be similar to the hidden layer but the inputs of the output layer are the outputs of the hidden layer. If there are two classes and a hundred neurons in the hidden layer then the weight matrix of the output layer is to be a 1×100 matrix. The output corresponds with a probability that a domain is a shadowed domain.

[0154] The neural network **550** is one example, and other neural networks can be used to detect shadowed domains. The other neural networks can include more layers than in neural network **550** and can have different layers including recurrent neural network layers, convolutional neural network layers, Long Short-Term Memory (LSTM) layers, gated recurrent layers, transformer

layers, dropout layers, etc. The other neural networks can use various embeddings, regularization, multi-representation, different activation functions, different sizes for each layer, etc.

[0155] FIG. **6**A is a flow diagram illustrating an embodiment of a process for detecting shadowed domains. In some embodiments, the process **600** is implemented using the detection pipeline **170** of FIG. **1** and comprises:

[0156] In **610**, the detection pipeline collects new hostnames for a predetermined period of time.

[0157] In **620**, the detection pipeline selects candidate shadowed domains from the new hostnames.

[0158] In **630**, the detection pipeline performs classification of the candidate shadowed domains based on a plurality of features relating to the candidate shadowed domains to output a set of identified shadowed domains.

[0159] In **640**, optionally, the detection pipeline performs post-processing on the set of identified shadowed domains.

[0160] In **650**, the detection pipeline performs an action based on the set of identified shadowed domains.

[0161] Some of the benefits of process **600** include less computational resources used to determine whether a new hostname is a shadowed domain thus costs are decreased, calculations are performed faster, and less false positives occur as a result.

[0162] FIG. **6**B is a flow diagram illustrating an embodiment of a process for collecting new hostnames for a predetermined period of time. In some embodiments, the process **6000** is an implementation of operation **610** of FIG. **6**A and comprises:

[0163] In **6010**, the detection pipeline determines whether a newly observed hostname (NOH) from a new hostnames dataset is found in an allowlist.

[0164] In **6020**, in the event that the NOH is found in the allowlist, the detection pipeline determines that the NOH is not shadowed.

[0165] FIG. **7** is a flow diagram illustrating an embodiment of a process for performing post-processing on a set of identified shadowed domains. In some embodiments, the process **700** is an implementation of operation **640** of FIG. **6**A and comprises:

[0166] In **710**, the detection pipeline compares a subnetwork of an IP address of the identified shadowed domain with a subnetwork of an IP address of a root domain associated with an identified shadowed domain. In some embodiments, the detection pipeline compares a subnetwork of an IP address of the identified shadowed domain with a subnetwork of an IP address of a root domain associated with an identified shadowed domain using an active DNS data sources and a passive DNS data source.

[0167] In **720**, in response to a determination that the subnetwork of an IP address of the identified shadowed domain matches the subnetwork of an IP address of the root domain associated with the identified shadowed domain, the detection pipeline determines that the identified shadowed domain is likely benign.

[0168] FIG. **8** is a flow diagram illustrating another embodiment of a process for performing post-processing on a set of identified shadowed domains. In some embodiments, the process **800** is an implementation of operation **640** of FIG. **6**A and comprises:

[0169] In **810**, the detection pipeline compares a time since creation of an identified shadowed domain with a predefined threshold.

[0170] In **820**, in response to a determination that the time since creation of the identified shadowed domain is equal to or exceeds the predefined threshold, the detection pipeline determines that the identified shadowed domain is not shadowed.

[0171] FIG. **9** is a flow diagram illustrating yet another embodiment of a process for performing post-processing on a set of identified shadowed domains. In some embodiments, the process **900** is an implementation of operation **640** of FIG. **6**A and comprises:

[0172] In **910**, the detection pipeline compares a time since the root domain of an identified shadowed domain was registered with a predefined threshold.

[0173] In **920**, in response to a determination that the time since the root domain of the identified shadowed domain was registered is less than or equal to the predefined threshold, the detection pipeline determines that the identified shadowed domain is not shadowed.

[0174] In some embodiments, operation **640** of FIG. **6**A includes one or more of process **700** of FIG. **7**, process **800** of FIG. **8**, and/or process **900** of FIG. **9**.

[0175] Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

Claims

1. A system, comprising: a processor configured to: collect new hostnames for a predetermined period of time; select candidate shadowed domains from the new hostnames; perform classification of the candidate shadowed domains based on a plurality of features relating to the candidate shadowed domains to output a set of identified shadowed domains; perform post-processing on the set of identified shadowed domains based on a time associated with an identified shadowed domain; and perform an action based on the set of identified shadowed domains; and a memory coupled to the processor and configured to provide the processor with instructions.

2. The system of claim 1, wherein the collecting of the new hostnames for the predetermined period of time comprises to: determine whether a newly observed hostname (NOH) from a new hostnames dataset is found in an allowlist; and in the event that the NOH is found in the allowlist, determine that the NOH is not shadowed.

3. The system of claim 1, wherein the performing of the classification of the candidate shadowed domains comprises to: extract the plurality of features relating to the candidate shadowed domains; and perform the classification of the candidate shadowed domains using a model or a set of rules.

4. The system of claim 3, wherein the model is a machine learning model.

5. The system of claim 3, wherein the model is a machine learning model, the machine learning model being a neural network.

6. The system of claim 1, wherein the performing of the post-processing on the set of identified shadowed domains, comprising to: compare a subnetwork of an IP address of an identified shadowed domain with a subnetwork of an IP address of a root domain associated with the identified shadowed domain; and in response to a determination that the subnetwork of an IP address of the identified shadowed domain matches the subnetwork of an IP address of the root domain associated with the identified shadowed domain, determine that the identified shadowed domain is likely benign.

7. The system of claim 1, wherein the performing of the post-processing on the set of identified shadowed domains, comprising to: compare a time since creation of an identified shadowed domain with a predefined threshold; and in response to a determination that the time since creation of the identified shadowed domain is equal to or exceeds the predefined threshold, determine that the identified shadowed domain is not shadowed.

8. The system of claim 1, wherein the performing of the post-processing on the set of identified shadowed domains, comprising to: compare a time since a root domain of an identified shadowed domain was registered with a predefined threshold; and in response to a determination that the time since the root domain of the identified shadowed domain was registered is less than or equal to the predefined threshold, determine that the identified shadowed domain is not shadowed.

9. A method, comprising: collecting, using a processor, new hostnames for a predetermined period of time; selecting, using the processor, candidate shadowed domains from the new hostnames; performing, using the processor, classification of the candidate shadowed domains based on a plurality of features relating to the candidate shadowed domains to output a set of identified

shadowed domains; performing post-processing on the set of identified shadowed domains based on a time associated with an identified shadowed domain; and performing, using the processor, an action based on the set of identified shadowed domains.

10. The method of claim 9, wherein the performing of the classification of the candidate shadowed domains comprises: extracting the plurality of features relating to the candidate shadowed domains; and performing the classification of the candidate shadowed domains using a model or a set of rules.

11. The method of claim 10, wherein the model is a machine learning model.

12. The method of claim 10, wherein the model is a machine learning model, the machine learning model being a neural network.

13. The method of claim 9, wherein the performing of the post-processing on the set of identified shadowed domains, comprising: comparing a subnetwork of an IP address of the identified shadowed domain with a subnetwork of an IP address of a root domain associated with the identified shadowed domain; and in response to a determination that the subnetwork of an IP address of an identified shadowed domain matches the subnetwork of an IP address of the root domain associated with the identified shadowed domain, determining that the identified shadowed domain is benign.

14. The method of claim 9, wherein the performing of the post-processing on the set of identified shadowed domains, comprising: comparing a time since creation of the identified shadowed domain with a predefined threshold; and in response to a determination that the time since creation of an identified shadowed domain is equal to or exceeds the predefined threshold, determining that the identified shadowed domain is not shadowed.

15. The method of claim 9, wherein the performing of the post-processing on the set of identified shadowed domains, comprising: comparing a time since a root domain of an identified shadowed domain was registered with a predefined threshold; and in response to a determination that the time since the root domain of the identified shadowed domain was registered is less than or equal to the predefined threshold, determining that the identified shadowed domain is not shadowed.

16. A computer program product embodied in a non-transitory computer readable medium and comprising computer instructions for: collecting new hostnames for a predetermined period of time; selecting candidate shadowed domains from the new hostnames; performing classification of the candidate shadowed domains based on a plurality of features relating to the candidate shadowed domains to output a set of identified shadowed domains; performing post-processing on the set of identified shadowed domains based on a time associated with an identified shadowed domain; and performing an action based on the set of identified shadowed domains.

17. The computer program product of claim 16, wherein the performing of the classification of the candidate shadowed domains comprises: extracting the plurality of features relating to the candidate shadowed domains; and performing the classification of the candidate shadowed domains using a model or a set of rules.

18. The computer program product of claim 17, wherein the model is a machine learning model.

19. The computer program product of claim 16, wherein the performing of the post-processing on the set of identified shadowed domains, comprising: comparing a time since creation of the identified shadowed domain with a predefined threshold; and in response to a determination that the time since creation of an identified shadowed domain is equal to or exceeds the predefined threshold, determining that the identified shadowed domain is not shadowed.

20. The computer program product of claim 16, wherein the performing of the post- processing on the set of identified shadowed domains, comprising: comparing a time since a root domain of an identified shadowed domain was registered with a predefined threshold; and in response to a determination that the time since the root domain of the identified shadowed domain was registered is less than or equal to the predefined threshold, determining that the identified shadowed domain is not shadowed.