

COMPUTING DEVICE AND METHOD OF DETECTING COMPROMISED NETWORK DEVICES BASED ON DNS TUNNELING DETECTION

FIELD

[0001] This technical solution relates to the field of computing, in particular to the field of information security, namely, to a computer-implementable method and a computing device for detecting compromised devices based on DNS tunneling detection.

BACKGROUND

[0002] DNS tunneling is one of the most common methods of exchanging information with a compromised device over the Internet for intruders.

[0003] This term usually means a method of information exchange, where the message is transmitted by a standard DNS query under the guise of a domain name of the third or higher level, for example, *extremely_hidden_secret_message.domain.com*.

[0004] Typically, the message is being encrypted prior to sending therefore, a domain name common for DNS tunneling usually looks as follows:

*MV4HI4TFNVSWY6K7NBUWIZDFNZPXGZLDOJSXIX3NMVZXGYLHMU====
==.domain.com.*

[0005] The mere fact of DNS tunnel availability in the network infrastructure outbound traffic is usually interpreted as an indication that at least one device of this infrastructure is compromised.

[0006] Two approaches are used in the prior art to detect compromised network devices using DNS tunneling, which can be conventionally determined as based on traffic volume analysis and payload analysis.

[0007] In the first case, when analyzing the traffic volume, attention is paid to the total number of queries to a particular domain and this number is compared to the average values. Since DNS tunneling is characterized by atypically large volume of DNS traffic, namely, tens and hundreds of times higher than typical values.

[0008] Thus, the patent document US 10044748 B2 (Georgia Tech Research Corp.) discloses a method for detecting compromised devices in a network based on determining the frequency of DNS queries, that is, when the frequency of queries is suspicious, processing is performed to determine the number of DNS queries for a certain period, and to compare the collected data to determine whether the number of the third level domain queries exceeds the number of the second level domain queries.

[0009] There is also known a solution, disclosed in the document US 10587646 B2, wherein DNS traffic anomalies are detected based on a threshold value of the number of queries per unit time.

[0010] However, for the systems implementing such approaches, sessions with small exchange volume may go undetected, when, for example, malicious

software (malware) uses a DNS tunnel for secret sending the compromised user credentials to the command server, but not for downloading or uploading a file or other significant amount of information.

[0011] When analyzing the payload, attention is paid to anomalous queries, especially if they comprise atypical domain names, strange characters, etc. Most often, this approach uses various statistical techniques to detect unusually long domain names in queries.

[0012] As an example, the document US 10270744 B2 (Infoblox Inc.) discloses an analytical system that processes DNS data, while determining the text string entropy in the DNS data stream based on the character set distribution in the text string, determines lexical features based on human-readable characters and human non-readable characters, in the text string determines the percentile value of the distribution of N-gram scores from a text string in the DNS data stream, based on which templates are formed to detect compromised devices.

[0013] However, intruders are also familiar with this approach, and they may deliberately not use domain names of the maximum available length at DNS tunneling, that complicates detection.

[0014] Thus, the need to develop technical solutions that would enable to detect compromised devices of the network infrastructure even with small volumes of data exchange and at DNS tunneling with deliberately “harmless” domain name length becomes obvious.

[0015] The other technical solution is known, which has been chosen as the closest analogue of RU 2668710 C1 (Group-IB TDS), which is aimed at detection of malicious domain names in the network traffic. This document describes a heuristic approach of detecting malicious domain names. However, computing the domain name maliciousness using a combination of techniques is resource- and time-consuming. Moreover, the solution described herein enables not only to detect malicious domain name, but also to block the device associated with it.

[0016] Therefore, this technical solution is created to overcome at least one of the above-mentioned disadvantages of the known systems and methods of detecting compromised network devices using DNS tunneling.

SUMMARY

[0017] The technical problem to be solved by the claimed solution is implementation of a system and computer-implementable method of detecting compromised network devices based on DNS tunneling. Additional embodiments of this technique are presented in the dependent claims.

[0018] The technical result consists in improving the accuracy of detecting compromised network devices based on DNS tunneling.

[0019] The result claimed is achieved due to implementation of the computer-implementable method of detecting compromised network devices using DNS tunneling, comprising the steps where: receiving a traffic of a protected network infrastructure; extracting at least one DNS query from the traffic; filtering a list of extracted DNS queries by excluding from the list a queries to a predefined domain

name types; feeding one-by-one each of a plurality of domain names remaining in the list as an input to a pre-trained decision rule; if at least one domain name obtains a negative assessment of the decision rule, increasing a penalty for at least one network device sent at least one DNS query, containing this domain name; if the amount of penalties of at least one network device exceeds a preset threshold, this at least one network device is considered to be compromised, and displaying a corresponding notification.

[0020] In a particular embodiment of the method described, additionally to the traffic of the protected network infrastructure received inbound network traffic of the protected network infrastructure is also received.

[0021] In a particular embodiment of the method described the traffic is received by a traffic mirroring method.

[0022] In a particular embodiment of the method described for each DNS-query identified in a traffic at least the following data are extracted: domain name; sender IP address and recipient IP address.

[0023] In a particular embodiment of the method described a presence of at least one recipient IP address is checked in the black and whitelist, and in response to presence of at least one recipient IP address in the blacklist, the penalty for at least one network device which sent at least one DNS query containing this at least one recipient IP address, is increased; in response to a presence of at least one recipient IP address in the whitelist, the DNS query is skipped.

[0024] In a particular embodiment of the method described, the predetermined types of domain names include at least: second level domain names, low entropy domain names, and knowingly trusted names from well-known reliable sources.

[0025] There is another embodiment of the described method, wherein the pre-trained decision rule comprises at least one the following heuristic methods: dividing domains by entropy level, selecting n-grams in domain names.

[0026] In a particular embodiment of the described method, the pre-trained decision rule is a LSTM neural network.

[0027] In a particular embodiment of the described method, the LSTM neural network is additionally fed with statistics of network interactions of the network device which sent the DNS query, containing the given domain name, as well as the penalty of this network device, wherein statistics of network interactions is collected during last T minutes.

[0028] In a particular embodiment of the described method, the pre-trained decision rule is a combination of LSTM neural network and at least one of the following heuristic methods: dividing domains by entropy level, selecting n-grams in domain names.

[0029] In a particular embodiment of the described method, the method further comprising:

searching for at least one domain name obtained the negative assessment in the history of all inbound and outbound traffic of at least one network device of the protected infrastructure, and

in response to finding entries of the at least one domain name obtaining the negative assessment:

either additionally penalizing every network device of the protected infrastructure which communicated with the said domain name,

or penalizing every network device which is not penalized before.

[0030] In a particular embodiment of the described method, the method further comprising:

blocking the domain name obtained negative assessment for every network device included into the protected infrastructure,

generating notifications on maliciousness of the domain name blocked,

creating template emails for notifying interested parties about domain name blocked,

adding the domain name blocked to at least one blacklist.

[0031] The result claimed is also achieved due to creating a computing device configured to perform a method of detecting compromised network devices based on DNS tunneling.

DESCRIPTION OF THE DRAWINGS

[0032] Implementation of the technique will be further described in accordance with the attached drawings, which are presented to clarify the technique chief matter and by no means limit the field of the technique. The following drawings are attached to the application:

[0033] Fig. 1 illustrates a simplified embodiment of the system for detecting compromised network devices based on DNS tunneling.

[0034] Fig. 2 illustrates a flowchart of steps of embodiment of the described technique of detecting compromised network devices based on DNS tunneling.

[0035] Fig. 3 illustrates a simplified schematic diagram of one non-limiting embodiments of a computing device for implementation of this technical solution.

DETAILED DESCRIPTION

[0036] Numerous implementation details intended to ensure clear understanding of this technique are given in the detailed embodiment specification below. However, it is obvious to a person skilled in the art how to use this technique as with the given implementation details as without them. In other cases, the well-known methods, procedures and components have not been described in details so as not to obscure unnecessarily the present technique.

[0037] Besides, it will be clear from the given explanation that the technique is not limited to the given embodiment. Numerous possible modifications, changes, variations and replacements retaining the chief matter and form of this technique will be obvious to persons skilled in the art.

[0038] DNS tunneling is a technique that enables to transmit arbitrary traffic over the DNS protocol. In order to arrange a DNS tunnel, someone is required to receive it outside (i.e., at the point where the tunnel is directed to). The receiving point is the name server for a particular zone. From the inside the network name queries are made for this zone; this is the way the traffic is transmitted from the

inside the network. As a result of transformation, response data are returned - this is the way the traffic is transmitted inside the network. DNS tunneling cannot be prevented by simple firewall rules while allowing other DNS traffic. This is due to the fact that DNS tunnel traffic and legitimate DNS queries are indistinguishable.

[0039] Protected network infrastructure is a set of special equipment, software and user devices, which in the text of this application could be equally denoted by a general term for a given set of network devices, in respect to which the fact of potential compromise is actively monitored.

[0040] The term “penalty” in this specification means a cumulative assessment of the device maliciousness and/or compromise for a certain period of time.

[0041] This technique is intended to provide the computer-implementable method and system of detecting compromised network devices based on DNS tunneling detection.

[0042] The computer-implementable method of detecting compromised network devices claimed based on DNS tunneling detection is carried out using a system for detecting compromised network devices based on DNS tunneling detection shown in Fig. 1, which comprises at least: the protected network infrastructure 110, communication network 120, computing device for detection of compromised network devices based on DNS tunneling detection 130.

[0043] The computing device 130 is also communicatively coupled with traffic capture devices 112 to receive from them the network traffic retrieved from

the data transmission network 120, and is configured to further process the received network traffic to detect malicious domain names used for connection between an infected workstation or an infected computer and the managing command center with subsequent receiving control commands from it to execute a set of programmed malicious actions.

[0044] Accordingly, the network infrastructure protected comprises at least one user workstation 111, a network switch 112, which is configured to transmit (relay) all traffic of the network infrastructure 110.

[0045] In the system 100, shown in Fig. 1, a traffic capture device 112 is connected to a computing device by wire, for example, a network cable, or wirelessly, and is at least one of the following known network devices for capturing and transmitting network traffic: L2 level network switches using the technique of traffic mirroring of necessary network segments, such as, for example, SPAN traffic mirroring technique implemented in CISCO network equipment, network transparency tools, also called Security Delivery Platforms or Network Packet Brokers (NPB), and Test Access Points of various types, and also ICAP protocol-supported proxy servers operating within an established TCP connection, SMTP protocol-supported mail servers, etc.

[0046] In one embodiment of the present technique, the computing device 130 could be included directly in the data transmission network 120 or connected directly to the data transmission network 120. The computing device 130 being configured to extract or capture the network traffic from data transmission network 120 for

further analysis and processing to detect malicious domain names in the network traffic specified. In other words, in such an embodiment, the computing device 130 could have the functionality of the above-described traffic capture devices 112, while the computing device 130, if necessary, could use a secure data transmission/reception link to receive network traffic retrieved from the data transmission network 120.

[0047] In the other additional embodiment of the present technique, the network traffic capture devices described above 112 could be embedded or integrated into the computing device 130.

[0048] In the system 100 shown in Fig. 1, the computing device 130 is actually a software and hardware system and comprises a traffic capture module 131, communication module 132, filtering module 133, analysis module 134, decision module 135 and local data storage 136, each being connected to a communication bus 137, while each of the traffic capture module 131, communication module 132, filtering module 133, analysis module 134 and decision module 135 is configured to exchange data with the local data storage 136 via the communication bus 137, where the traffic capture module 131 is configured to exchange data via the communication bus 137, and the communication module 132 is also configured to exchange data via the communication bus 137 with the filtering module 133, which in turn, is configured to exchange data via the communication bus 137 with the analysis module 134, and the analysis module 134 is communicatively connected with the decision module 135 via the communication bus 137.

[0049] The local data storage 136 is intended for storing executable program instructions, which could control operation of the traffic capture module 131, communication module 132, filtering module 133, analysis module 134 and decision module 135. Besides, the local data storage 136 is used in the computing device 130 for storing different data used at operation of the computing device 130.

[0050] In particular, the local data storage 136 stores data on the analytic reports available, each of them being associated with a specific domain name, and each of them contains data on the numerical values of penalties assigned to the specific domain name by the analysis module 134, and also data related to the decision taken by the decision module 135 in respect to the network device, from which the DNS query has been sent, comprising this domain name.

[0051] The local data storage 136 also stores data on known domain names, blacklists and whitelists of domain names. It is also worth noting that information about blacklists and whitelists in the additional embodiment could be received from the external trusted databases (not shown) via the communication network 120.

[0052] In additional embodiments of this technique the local data storage 136 also comprises data on each of the network devices 111 (for example, the user ones) in the protected infrastructure 110. The data described include at least: statistics of network interactions of a specific device 111 in the protected infrastructure 110 within T minutes, where the numerical value T is preselected, this network device penalties, and notes on the fact of this device 111 compromise.

[0053] In some embodiments of this technique at least some of the above data stored in the local data storage 136 could be stored in a cloud storage (not shown), where these data could be stored as on a single remote file server included in the data transmission network 120 or in the other data transmission network different from the data transmission network 120, as on a plurality of remote file servers distributed in the data transmission network 120 or in the other data transmission network different from the data transmission network 120. The computing device 130 could be configured to communicate for data exchange, in particular, to establish a secure data transmission/reception link with such cloud storage using wire and/or wireless connection method. In the embodiment, wherein the computing device 130, included in the data transmission network 120, is connected with the cloud storage via at least one other data transmission network different from the data transmission network 120, for connection of these different data transmission networks there could be used at least one special switching device, such as network hubs, network routers and other known switching devices.

[0054] In one of the embodiments of this technique, the computing device 130 could comprise a separate local analytic report storage (not shown) intended for storing data on available analytic reports. Such a separate local analytic report storage could be connected, via the communication bus 137, to the filtering module 133, enabling the filtering module 133 to access this local analytic report storage to obtain data on available analytic reports from it, and also be connected via the communication bus 137, with the decision module 135 being able to access such a

local analytic report storage for writing a new analytic report to it, what is described in more detail below.

[0055] In one of the embodiments the filtering module 133, analysis module 134 and decision module 135 could be implemented as a single processor, such as a general-purpose processor or a special-purpose processor (e.g., digital signal processors, ASICs, and etc.). Thus, the processor implementing the said modules 131-135 in the computing device 130 could be configured to execute program instructions stored in the local data storage 136 implementing the functionality of the traffic capture module 131 for receiving the traffic from the protected infrastructure 110, functionality of the filtering module 133 for extracting domain name data from data packets in the received network traffic, functionality of the analysis module 134 for determining the numerical values of suspiciousness features for the analyzed domain name, and functionality of the decision module 135 for taking a final decision regarding the maliciousness of a certain device using the obtained set of suspiciousness features with the numerical values assigned to them.

[0056] In one of the embodiments of this technique each of the filtering module 133, analysis module 134 and decision module 135 could be implemented as at least one separate processor configured to implement specific functions of the modules.

[0057] In the computing device 130 of the system 100 shown in Fig. 1, the communication module 132 is wired, for example, using a coaxial cable, twisted

pair, fiber optic cable, or other physical connection, to the traffic capture device (112 or 131) to receive network traffic from them.

[0058] Thus, the communication module 132 is configured to connect to one or more of the above traffic capture devices (112 or 131) included in the data transmission network 120, and to receive network traffic retrieved by the said traffic capture devices 112 from the data transmission network 120 from the specified one or more traffic capture devices (112 or 131). In the other embodiment of this technique, the communication module 132 could be wirelessly connected to the traffic capture devices (112 or 113), for example, using WiFi-based communication line, 3G-based communication line, LTE-based communication line, etc.

[0059] The communication module 132 in the computing device 130 according to this technique could be implemented in the form of a network adapter equipped with necessary connectors for connecting physical cables of the required types to them, depending on the types of physical connections used for communication with the capture traffic devices (112 or 131). In one of the embodiments of the present technique, the communication module 132 in the computing device 130 could be implemented as a WiFi adapter, 3G adapter, LTE adapter or other wireless adapter, depending on the type of wireless communication line used for communication with any external network devices and remote data storages that could establish a wireless communication link with a computing device to exchange the necessary data with it. Thus, the communication module 132 in the computing device 130 could be configured to receive input data from one or more

network devices by wire and/or wirelessly, and also could be configured to send output data to the other network devices by wire and/or wirelessly.

[0060] The communication module 132 could also be any known communication device, such as a transceiver, modem and/or network interface card for communicating with external network devices of any type via a wired or wireless communication network, for example, using Ethernet connection, digital subscriber line (DSL), telephone line, coaxial cable, cellular telephone system, etc. In some embodiments of the present technique, the network traffic received by the communications module 132 could be at least temporarily stored in the local data storage 136. In the other embodiments of the present technique, the network traffic received by the communication module 132 could be at least temporarily stored in a separate local network traffic storage (not shown), different from the local data storage 136 and connected via the communication bus 137 to the communication module 132. In the other embodiments of the present technique, the network traffic received by the communication module 132 could be at least temporarily stored in a separate remote network traffic storage (not shown), communicatively connected with the communication module 132 by wire and/or wirelessly.

[0061] In the computing device 130 of the system 100 shown in Fig. 1, the filtering module 133 is configured to connect or communicate, via the communications bus 137, to the communication module 132 so as to receive the captured network traffic therefrom. The filtering module 133 extracts a plurality of data packets, comprising DNS queries from the network traffic received, using, for

example, known techniques for extracting DNS queries in data packets, such as zeek, snort, Suricata, Bro, Kuko, and the like techniques, including the use of improved techniques based on at least one of the above techniques. The filtering module 133 is also configured to analyze each of the plurality of the extracted data packets, so as to extract at least one domain name and sender and/or recipient IP addresses therefrom.

[0062] In an embodiment of the present technique, wherein the network traffic received is stored in the local data storage 136, the filtering module 133 could be configured to access the local data storage 136 or communicate with it via the communication bus 137, so as to extract the stored network traffic therefrom for subsequent retrieval of a plurality of data packets therefrom, as described above.

[0063] In an embodiment of the present technique, wherein the received network traffic is stored in a separate local network traffic storage (not shown), the filtering module 133 could be configured to access such local network traffic storage or communicate with it via the communication bus 137, so as to extract the stored network traffic therefrom for subsequent retrieval of a plurality of data packets therefrom, as described above. In an embodiment of the present technique, wherein the received network traffic is stored in a separate remote network traffic storage (not shown), the filtering module 133 could be configured to access such remote network traffic storage or communicate with it using the communication module 132 connected to the filtering module 133 via the communication bus 137, so as to

extract the stored network traffic therefrom for subsequent retrieval of a plurality of data packets therefrom, as described above.

[0064] In one of the embodiments of the present technique after extracting domain names from the analyzed DNS queries the filtering module 133 additionally determines whether there is an analytic report for each of these domain names extracted.

[0065] In order to detect the availability of an analytical report already generated for a given domain name, the filtering module 133 performs at least the following operations: accessing the local data storage 136 or communicating with it via the communication bus 137, so as to receive data on the available analytical reports therefrom, where each of the reports is associated with a specific domain name, for which this analytical report has been generated; setting up a correspondence of each of the domain names extracted to one of the available analytic reports by character-by-character comparison of each of the extracted domain names with the domain names identifying the analytic reports in the specified received data. Thus, the filtering module 133 compares each domain name extracted with the domain names identifying the analytic reports in the analytic report data received, where a match of the domain name extracted with a domain name, identifying one of the available analytic reports, means that there is already an analytic report stored in the local data storage 136 for the extracted domain name, that is indicative of the fact that the domain name extracted has already been previously analyzed in the computing device 130.

[0066] In an embodiment, wherein the analytic report data are stored in a separate local analytic report storage (not shown), the filtering module 133 accesses such local analytic report storage or communicates with it using the communication module 132 to obtain the data on the available analytic reports therefrom, where each of the reports is associated with a specific domain name, for which this analytic report has been generated, for subsequent setting up a correspondence of the domain name extracted to one of the available analytic reports.

[0067] Thus, the filtering module 133 enables filtering at least part of the domain names extracted from the data packets of the received network traffic using the minimum computing resources of the computing device 130, which ultimately significantly speeds up the process of detection of malicious domain names in the network traffic.

[0068] Further, the filtering module 133 transmits to the analysis module 134 only those extracted domain names, for which there is no analytic report in the local data storage 136, that is, only those domain names that have not been previously analyzed in the computing device 130, in particular in its decision module 135.

[0069] In the computing device 130 of the system 100 shown in Fig. 1, the analysis module 134 is configured to connect to the filtering module 133 or communicate with it via the communication bus 137, so as to obtain information therefrom, for example, information about domain names for which the filtering module 133 has detected the absence of an analytical report, as described above.

[0070] In an embodiment of the present technique, wherein the domain names extracted by the filtering module 133 are stored in the local data storage 136, the analysis module 134 accesses the local data storage 136 or communicates with it via the communication bus 137, so as to obtain the data on domain names therefrom for further analysis in this analysis module 134, as described below.

[0071] In the other embodiment of the present technique, wherein the domain names extracted by the filtering module 133 are stored in a separate local data storage, the analysis module 134 accesses such local data storage or communicates with it via the communication bus 137 or via wireless communication link through the communication module 132, so as to obtain the data on domain names therefrom for further analysis in this analysis module 134, as described below.

[0072] Analysis module 134 analyzes each of the obtained domain names using a predetermined set of techniques or methods for analyzing suspiciousness, with assigning a predetermined penalty numerical value to each of the domain names obtained.

[0073] In particular, the analysis module 134 is based on a decision rule, on the basis of which a verdict is rendered with regard to maliciousness or safety of a particular domain name with use of a set of suspiciousness features of this domain name, and is indicative of the potential maliciousness of the device (the user device 111) for the decision module 135. Both heuristic and statistical algorithms, including machine learning algorithms (MLA), could act as a decision rule. An example of a heuristic rule could be a division of domains by the entropy level so that domains

with high level of entropy are recognized as suspicious, that is, domain names, which entropy level exceeds a predetermined empirically derived threshold, are recognized as suspicious. Approaches based on heuristic methods, for example, such as selecting n-grams and subsequent computing the domain name entropy, are also possible.

[0074] One of the embodiments is based on a neural LSTM network, where dimensionality of the feature space is normalized and reduced on the first layer by a well-known method, the next is the first learning layer, LSTM as such with empirically derived coefficients. The next layer clears the results by a well-known method. After that, there are two additional intermediate layers, and on the last one a binary verdict is rendered, whether the domain is suspicious.

[0075] In one of the embodiments, the analysis module 134 could be based on a set of decision rules, where each of the rules makes its own decision or renders its own verdict with regard to maliciousness or safety of a particular domain name using one and the same set of suspiciousness features of this domain name. In one of the variants of this embodiment of the present technique, the final verdict with regard to the domain name safety could be rendered by the analysis module 134. For example, in case, where at least one of a predetermined set of decision rules or all of the predetermined set of decision rules have rendered a verdict with regard to the domain name safety, otherwise, where none of the predetermined set of decision rules has rendered a verdict with regard to the domain name safety, the module 134 could render the final verdict with regard to the domain name maliciousness. In the other

variant of this embodiment, the analysis module 134 could render its final verdict with regard to maliciousness or safety of a domain name by comparing the total weight of personal verdicts accepted by a predetermined set of decision rules, where a predefined weighting factor is automatically assigned to any personal verdict rendered by each decision rule from the predetermined set. Thus, the analysis module 134 renders one or another final verdict taking into account all intermediate verdicts having one or another weight in the total weight taken into account for rendering the final verdict.

[0076] In one of the embodiments of the present technique the analysis module 134 could be equipped with a special framework (software platform) that enables, for example, an operator or user of the computing device 130 to change a predetermined set of techniques for domain name maliciousness analysis used by the analysis module 134 for obtaining a verdict on domain name maliciousness corresponding to one of the analysis techniques from the predetermined set, for example, selection of a group of heuristic techniques described below or machine learning algorithms, that enhances the functionality of as the analysis module 134 individually, as the computing device 130 as a whole. Besides, this framework serves to obtain data on the T time value, which determines the statistics of interaction of the particular device 111 and the protected infrastructure 110.

[0077] After rendering the final verdict, the analysis module 134 prepares or generates a separate analytic report for each domain name under investigation, where such an analytic report at least comprises all the data from the corresponding log file

created during the analysis for the specified domain name, and also the data on the final verdict rendered by the analysis module 134 with respect to the domain name, and also the data on the decision rule used by the analysis module 134. In an embodiment of the present technique, wherein several decision rules are used, the analytic report prepared or generated by the analysis module 134 could also comprise the data on each of the decision rules used for rendering the final verdict, and the data on the verdicts of each of the decision rules with regard to maliciousness or safety of the domain name.

[0078] In some embodiments of the present technique, the analysis module 134 could be divided into several independent modules, where each of the modules could perform at least one of the above-described functionalities inherent in the analysis module 134, and which are configured to communicate with each other and the other structural modules of the computing device 130 via the communication bus 137. In this case, one of such independent modules could be configured to create at least one log file, or there could be, for example, an additional module for creating log files, configured to communicate with the said independent modules via, for example, the communication bus 137, so as to receive from each of them, for example, the data on a specific suspiciousness feature with a numerical value assigned to it for the same domain name.

[0079] For the purpose of storing the prepared or generated analytic report in the local data storage 136, the analysis module 134 accesses the local data storage 136 or communicates with it via the communication bus 137, so as to transmit the

said analytic report to the local data storage 136 for storing thereof. Besides, an embodiment is possible, where the prepared or generated analytic report is stored in a separate local analytic report storage (not shown).

[0080] In the computing device 130 of the system 100 shown in Fig. 1, the decision module 135 is configured to connect to the analysis module 134 or to communicate with it via the communication bus 137, so as to receive the log files, prepared or generated by the analysis module 134, therefrom.

[0081] The decision module 135 analyzes the penalties associated with each domain name with the numerical values assigned to it, comprised in each of the log files received using a predetermined set of analysis rules, assigning each domain name, for which a corresponding log file has been created, to the malicious domain names, if the obtained results of the suspiciousness features analysis are typical of the malicious domain names, or to the trusted domain names, if the results of the suspiciousness features analysis are typical of the trusted domain names. Thus, the decision module 135 takes the final decision or renders the final verdict regarding the device compromise based on the assessment of the domain name maliciousness or safety using the log file created by the analysis module 134 for each domain name extracted.

[0082] It is worth noting that the communication module 132, filtering module 133, analysis module 134 and decision module 135 perform their above functions actually in real time, that is, the communication module 132 continues to receive the network traffic, the filtering module 133 performs its operations filtering at least a

part of the DNS queries extracted from the data packets in the network traffic received based on predetermined types of domain names, the analysis module 134 performs its operations analyzing domain names and assigning to them the suspiciousness features - penalty values with the subsequent creation of a log file for each domain name individually, and the decision module 135 performs its operations analyzing at least part of the log files received from the analysis module 134, and storing at least part of the generated or prepared analytic reports in the local data storage 136.

[0083] In one of the embodiments of the present technique, the above functional modules could be separated or combined while maintaining the functions inherent in each module individually. For example, the analysis module 134 and the decision module 135 could be combined, for example, into one module for detecting malicious domain names and, on the basis of them, detecting the compromised network devices 111 of the protected infrastructure, which has all the above functionalities inherent in each of them individually. Alternatively, in the other embodiment of the present technique, the filtering module 133 and the analysis module 134 could be combined, for example, into one module having all of the above functionalities inherent in each of them individually.

[0084] In the other embodiment of the present technique, the communication module 132 could be divided into several separate communication modules, where each of them provides at least one of the known methods of wired and/or wireless communication in the computing device 130.

[0085] In accordance with one more aspect, a method for detecting the compromised network devices using DNS tunneling 200 is proposed. It is illustrated as the simplified flow chart in Fig. 2.

STEP 210 RECEIVING TRAFFIC OF THE PROTECTED NETWORK INFRASTRUCTURE

[0086] At the first step of the technique described, the traffic of the network infrastructure protected 110 is received on the device 130 via the data transmission network 120.

[0087] The operations at this implementation step of the method claimed 200 include connection, via the communication module 132, to at least one of the network traffic capture devices included in the data transmission network (112 or 131).

[0088] In this case, the traffic of the protected network infrastructure could mean both outbound traffic of the protected network infrastructure, and additionally inbound network traffic.

[0089] It is worth noting that traffic could be received by any method known in the art, for example, by a traffic mirroring method.

[0090] By way of example, but not limited to, the traffic mirroring is executed by duplicating the packets transmitted through one port of network switch 112 to other one to be received by the computing device 130. It is worth noting that most of modern network switches enable to duplicate traffic using built-in functions. Besides, it is additionally possible to use, for example, tcpdump console utility,

which is included in most Unix systems and enables to intercept and display network traffic by the traffic capture module 131.

[0091] Once the traffic is received from the protected network infrastructure 110, it is stored in the database 136 of the computing device 130.

[0092] Then, the present technique proceeds to analysis of the traffic received by the methods described below.

STEP 220 EXTRACTING AT LEAST ONE DNS QUERY FROM THE TRAFFIC

[0093] Analysis of the traffic obtained at the previous step begins at step 220, where at least one DNS request is extracted therefrom.

[0094] Extraction of at least one DNS query from the traffic could be performed by the computing device 130 via the filtering module 133 based on any technique known in the art, for example, using a specially written parser or network traffic analysis services.

[0095] As an example of the embodiment of the present technique, but not limited to, the following tools could be used to detect DNS queries: zeek, snort, Suricata, etc. However, it is worth noting that the use of such tools enables to get a full picture of events on the network according to all the most popular protocols: HTTP, DNS, TLS, SMB, FTP, RDP, SSH, ICMP, etc., including industrial protocols s7, iec104, opcua, etc.

[0096] In one of the additional embodiments of the described technique, extracting DNS query also comprises extraction from the DNS query, using the

filtering module 133, at least the following data: domain name; sender IP address and recipient IP address.

[0097] Additionally, at the described step of the method 200, there could be a step of checking for availability of an analytic report associated with the domain names under investigation in the local data storage 136. It could include execution, using the filtering module 133, at least the following substeps: (i) obtaining data on analytic reports; (ii) setting up a correspondence of each of the domain names extracted to one of the available analytic reports by character-by-character comparison of each of the domain names extracted with the domain names identifying the analytic reports in the specified received data. In some embodiments of the present technique, in response to the fact that there is no analytic report for the extracted domain name, according to the proposed method, the specified domain name is transmitted via the filtering module to the analysis module 134.

[0098] If the filtering module 133 extracts an analytic report that is indicative of the domain name legitimacy, then all DNS queries comprising the domain name specified in the report are filtered (excluded from the further analysis). Otherwise, if the filtering module 133 extracts an analytic report comprising data on the malicious domain name, then, there the search in the traffic of the protected infrastructure 110 for all network devices 111 interacting with the identified domain name and computing a penalty for the identified network devices 111 is proceeded in accordance with the rules described hereafter (step 250).

[0099] Then, all the queries identified in the traffic are stored in the local data storage 136 of the system 100 in the form of a list comprising a DNS query and the corresponding domain name; sender IP address and recipient IP address.

STEP 230 FILTERING A LIST OF THE DNS QUERIES EXTRACTED, BY EXCLUDING FROM THE LIST A QUERIES TO A PREDEFINED DOMAIN NAME TYPES

[0100] Then, during the second step of analysis, the list stored in the local data storage 136 is filtered by the filtering module 133. It is worth noting that filtering all types of queries to domain names is performed on the basis of well-known methods, and this filtering is based on the features of the domain name which the query is directed to.

[0101] The domain name types that are filtered, that is, excluded from the previously generated list, comprise at least the following: second level domain names, low entropy domain names, and knowingly trusted names from well-known reliable sources.

[0102] Thus, the second level domain names are additionally excluded from the list, stored in the local data storage 136, using the filtering module 134. In this case, at least the following operations are executed: (i) communicating with a data storage, including a separate local 136 or remote trusted domain name storage (not shown) depending on the embodiment, as described above to obtain a list of domain names therefrom, (ii) transforming each domain name into a string and checking the number of dots in this string (iii) depending on whether the number of dots in the

string exceeds one (which corresponds to a second level domain name), the domain name is either filtered out and tagged as a trusted domain name, or remains in the list.

[0103] It is worth noting that the domain name entropy could be referred to the features allowing to determine the use of DNS tunnels. Conventional domain names are generally composed of English words; such names have low entropy. At the same time, the encoded data have high entropy. It is well-known that the DNS tunneling tools encode the data put into the DNS query. Thus, domain names with high entropy could indicate the use of DNS tunneling. In view of this, at this step of the present technique it is proposed to filter out the domain names with low entropy.

[0104] Entropy of a domain name could be determined by any well-known method using the filtering module 133. By way of example, but not limited to, entropy could be determined by the method described below.

[0105] As an example of the present technique implementation in the filtering module for sorting domain names with low entropy, at least the following substeps could be executed: (i) determining the frequency of each of a plurality of predetermined N-grams occurrence in each of the domain names obtained, with each N-gram corresponding to a combination of N consecutive characters. N-grams could be obtained from the local data storage 136, or from a wirelessly connected data storage; alternatively, they could be specified by an operator in a specific interface of the computing device 130; (ii) determining the entropy of N-grams of the specified analyzed domain name as a function of the specific frequency of each N-

gram occurrence in this analyzed domain name. Actually, computing could be performed by any method known in the art; (iii) comparing the determined N-gram entropy of the analyzed domain name with a preset N-gram entropy threshold value, which determines domain names with low entropy; the threshold value is predetermined empirically; (iv) filtering out (excluding) domain names, which entropy is below the preset threshold value.

[0106] Also, knowingly trusted names previously obtained from well-known reliable sources are additionally excluded from the list stored in the local data storage 136 using the filtering module 134. In this case, at least the following operations are executed: (i) communicating with a data storage, including a separate local 136 or remote trusted domain name storage (not shown) depending on the embodiment, as described above to obtain a set of domain names therefrom, (ii) determining a match of each of the extracted domain names with one of the known domain names by character-by-character comparison of each of the extracted domain names with the domain names in the obtained set of domain names, and (iii) depending on the matching result, the domain name is either filtered out and tagged as a trusted domain name, or it remains.

[0107] Alternatively, in additional embodiments, it is a possible case when the step of filtering queries to knowingly trusted names from well-known reliable sources could be skipped and performed at a later step using the analysis module 134, being included information of the decision rule learning samples.

[0108] In some additional embodiments of the present technique, domain names could also be encoded by any well-known method to support the localized domain names, i.e., domain names that are composed not of Latin characters, but of other national alphabet characters, such as, for example, доменное.имя.на.русском.рф (that means domain.name.in.russian.rf).

[0109] In one of the embodiments of the present technique, if a domain name is found in the locally accessible database 132 in the system 100 or in a trusted database accessed over the network, then the method proceeds to the last step of the present technique, which is disclosed in detail below (step 260).

[0110] In one of the embodiments, at this step there is additional checking at least one extracted sender IP address and/or recipient IP address in well-known reliable sources and in whitelist and blacklist stored in the database 136. This could be done by the method described above for checking domain names. Where, depending on the matching result, the domain name associated with the sender IP address and/or recipient IP address identified is either filtered out and tagged as a trusted domain name, or proceeds to the next step, described below, or is blocked if it is found in the blacklist or in a database of, for example, web resources that knowingly belong to cybercriminals.

STEP 240 FEEDING ONE-BY-ONE EACH OF A PLURALITY OF DOMAIN NAMES REMAINING IN THE LIST AS AN INPUT TO A PRE-TRAINED DECISION RULE

[0111] In embodiments of the described technique, the operations executed at step 240 of the method 200 include analysis, by the analyzing module 134, of the obtained set of suspiciousness features of the analyzed domain name using at least one pretrained decision rule, which could be both heuristic and statistical algorithms, including machine learning algorithms.

[0112] In one of the described embodiments of the present technique, heuristic rules are used as a decision rule. Thus, an example of a heuristic rule could be a division of domains by the entropy level so that domains with high level of entropy are considered as suspicious. Moreover, the entropy threshold level is determined empirically. Approaches based on selection of N-grams and subsequent computing their entropy are also possible, and also comparison of the computed entropy with a preset threshold value.

[0113] Besides, embodiments of the present technique are possible, wherein several heuristic approaches are used in parallel and their results are brought to a single binary assessment, that is, the decision could be one of two options: domain name is malicious or not. In this case, the analysis module 134 is configured to assign a reliability coefficient to each heuristic rule used in the process of obtaining the final assessment and to derive a single assessment of a domain suspiciousness based on these coefficients, while the analysis module 134 is configured to implement any known heuristic algorithms for determining the domain name maliciousness, for example, machine learning algorithms such as neural networks, decision trees, boosting algorithms, and also signatures.

[0114] In one of the described embodiments of the present technique, a pre-trained neural network could be a decision rule.

[0115] By way of illustration, but not limited to, an LSTM neural network could be used as a decision rule based on a neural network. LSTMs are long short-term memory networks (a representative of the recurrent network family, RNNs), they enable to analyze dependencies in a data sequence.

[0116] Further, the neural network designed uses vector representations of the input domain name prefix symbols as features for learning. In this embodiment, a prefix is considered to be a part of a domain made up of subdomains of the third and higher levels. For example, for `aaaa.drive.google.com` domain the prefix is the “aaaa.drive” string.

[0117] In embodiments of the present technique, domain name prefix could be determined by any well-known method. Thus, by way of illustration, a specially written script could be used, at implementation of which, firstly a generic top-level domain (gTLD) and/or a country code top-level domain (ccTLD), which are generic top-level domains, are selected using well-known methods. For example, in the URL `https://www.xyz.example.com/doc.html` the `example.com` string is a second level domain and `.com` is a generic top-level domain (gTLD). Likewise, in `https://www.example.ac.ru/` the `example.ac.ru` is a generic second level domain, and `.ru` is a country code TLD (ccTLD). Then, a subdomain of a lower level is selected and also excluded from the prefix.

[0118] The above mentioned operations, by way of example, could be implemented by a script written in Python using the 'tldextract' library, which uses well-known domain suffix references, including the PUBLIC SUFFIX LIST (<https://publicsuffix.org/>). A simplified version of the obtained results of the specified script execution is presented below:

```
tldextract.extract('http://forums.news.cnn.com/')
```

```
ExtractResult (subdomain='forums.news', domain='cnn', suffix='com').
```

[0119] It is worth noting that in addition to the domain name, the networking statistics collected during last T minutes of the network device which sent the DNS query comprising this domain name, and this device penalty are additionally fed to the decision rule input.

[0120] In the context of this specification, the networking statistics mean the number of interactions according to the above protocols for the last T minutes, and for the protocols above TCP this is the number of sessions, and for UDP protocols this is the number of datagrams.

[0121] In different embodiments, the number of T minutes could be manually set by an expert or user of the computing device 130, or could be set automatically, for example, depending on the number of network devices in the protected infrastructure 110.

[0122] It is worth noting that the built-in neural network can function with zero statistics parameters also, however, T is assumed to be small, i.e., by way of

example, but not limited to, 10 minutes, respectively, these statistics will appear quickly.

[0123] As for the penalty of the user device 111, its value is initially set to zero, which changes with the number of iterations in the system 100 by a method described in more detail below. In this case, all intermediate values of penalties are stored in the local data storage 136.

[0124] In embodiments of the present technique, the learning samples could be generated both in an isolated stand and in “field” conditions, within an operating and high-load network using well-known methods for building DNS tunnels. For example, such methods include the use of special utilities, such as `_NameServer` (NSTX), `Dns2tcp`, `Iodine`, `dnscat2`, etc.

[0125] Hereafter, general points of neural network building are described by way of example. Dimensionality of the feature space is normalized on the first layer by a well-known method. As an example of the procedure for normalizing the feature space, there could be the fact that all identified domains are added to the length of 256 with special characters on the right.

[0126] After this layer the next one is the first learning layer, LSTM as such with empirically derived coefficients. These coefficients are formed directly at the learning step, when tagged samples of domain names are loaded into the neural network, i.e. those for which the verdict is known in advance, i.e. the fact of their belonging to malicious or non-malicious domain names. It is worth noting that the said samples could be obtained both from external trusted sources and by empirical

method, for example, by manual activation of “maliciousness” flag for malicious domain names. Besides, the above data are also enriched with legitimate tagged domain names. That is, the analysis module 134 based on the LSTM neural network receives at the preparatory step both tagged malicious and legitimate domain names, on the basis of which the LSTM neural network generates coefficients, which are further used in the classification (prediction) mode.

[0127] The next layer of the LSTM neural network clears the results of the previous layer operation to obtain the most reliable results. There are many cleaning methodologies in the prior art, that could be implemented for this technique, one of the examples of such methods is discarding a part of extreme (maximum and minimum) values obtained during the analysis, for example, 25% per both.

[0128] This is followed by two additional intermediate layers that enable the LSTM neural network to select longer dependency chains as features. It has been empirically established that this number of inner layers gives the required balance of operation speed and errors. On the last layer of the neural network a binary verdict is rendered, whether a domain is malicious.

[0129] Domain name positive or negative assessment, obtained in the analysis module 134, is stored in the local data storage 136.

STEP 250 IF AT LEAST ONE DOMAIN NAME OBTAINS A NEGATIVE ASSESSMENT OF THE DECISION RULE, INCREASING A PENALTY FOR AT LEAST ONE NETWORK DEVICE SENT AT LEAST ONE DNS QUERY CONTAINING THIS DOMAIN NAME

[0130] Then, the verdict from the analysis module 134 is transmitted to the decision module 135, where, based on the parameters described hereafter, a decision is taken as to whether a network device has been compromised and the maliciousness degree thereof.

[0131] Thus, the decision module 135 checks the assessment of the decision rule included in the analysis module 134, and if the domain name has obtained a negative assessment of the decision rule, increases the amount of the penalty stored in the local data storage 136 for the network device from which the DNS request comprising this examined domain name has been sent.

[0132] In one of the additional embodiments, the decision module 135 additionally searches for a specific domain name in the history of all inbound and outbound traffic received from the protected infrastructure 110. Moreover, it is worth noting that not only DNS queries are being considered. If the decision module finds other entries of a specific domain name, then all network devices 111 of the protected infrastructure 110 communicating with this domain name are either additionally penalized or penalized if they have not been penalized during the last detection.

[0133] In one of the additional embodiments, the decision module 135 is additionally configured to block the malicious domain name detected for the entire protected infrastructure 110, to generate notifications (alerts), notifying the interested parties of an incident or display a corresponding notifications, to add the

malicious domain name detected to the blacklists accessible for the computing device 130 over the communication network, etc.

[0134] In one of the additional embodiments, the decision module 135 is additionally configured to extract from the local data storage 136 at least one sender IP address and/or recipient IP address associated with the malicious domain name, to identify in the local data storage 136 at least one additional domain name located at the same extracted IP address, and also to identify all the network devices 111 of the protected infrastructure 110 communicating with at least one additional identified domain name and to add a penalty to these network devices.

[0135] It is worth noting that the decision module also updates the blacklist (a list of malicious domain names), which is used at the filtering step in the filtering module 133 to reduce processing time and render a verdict on a similar query immediately.

[0136] Besides, the decision module processes the obtained penalty values of each device 111 and takes a decision of compromise thereof.

STEP 260 IF THE AMOUNT OF PENALTIES OF AT LEAST ONE NETWORK DEVICE EXCEEDS A PRESET THRESHOLD, THIS AT LEAST ONE NETWORK DEVICE IS CONSIDERED TO BE COMPROMISED, AND DISPLAYING A CORRESPONDING NOTIFICATION.

[0137] The decision module 135 is configured to check whether the amount of penalties of each network device (device 111) exceeds a preset threshold, consider

it a compromised network device in case of exceeding the threshold, and to display notifications, messages, alerts, etc. about this incident.

[0138] The threshold value of penalties is selected empirically, based on statistics of device compromise, frequency of interaction with malicious domain names, and so on.

[0139] In one of the additional embodiments, the decision module 135 is additionally configured to detect all contacts (interactions) of the detected compromised network device 111 inside the protected infrastructure 110. Moreover, special attention is paid to the remote-control protocols. In case of a large number of critical operations, for example, such as remote application launching, accessing the shared resources, etc., when interacting with any other network device 111, it is also penalized and then checked for penalty exceeding the compromise threshold value.

[0140] It is also worth noting that implementation of the above method 200 is also possible in specific networks. The technique claimed is especially effective in the networks with strict security policies. For example, in a network of video cameras, interactions over all network protocols, except for the protocol required for video transmission, would most likely be disabled. DNS protocol is required to provide routing on a network and is generally allowed even on such networks. In this case, it is highly likely that malefactors will use namely DNS as transport for data exfiltration.

[0141] Fig. 3 hereafter illustrates a schematic diagram of the computer device 130 processing the data required for embodiment of the claimed solution.

[0142] In general, the device 300 comprises such components as: one or more processors 301, at least one memory 302, data storage means 303, input/output interfaces 304, input/output means 305, networking means 306.

[0143] The device processor 301 executes main computing operations, required for functioning of the device 300 or functionality of one or more components thereof. The processor 301 runs the required machine-readable commands, contained in the random-access memory 302.

[0144] The memory 302, typically, is in the form of RAM and comprises the necessary program logic ensuring the required functional.

[0145] The data storage means 303 could be in the form of HDD, SSD, RAID, networked storage, flash-memory, optical drives (CD, DVD, MD, Blue-Ray disks), etc.

[0146] Interfaces 304 are standard means for connection and operation with server side, e.g. USB, RS232, RJ45, LPT, COM, HDMI, PS/2, Lightning, FireWire, etc. Selection of interfaces 304 depends on the specific device 300, which could be a personal computer, mainframe, server cluster, thin client, smartphone, laptop, etc.

[0147] As input/output means 305 there could be used: keyboard, joystick, display (touch-screen display), projector, touch pad, mouse, trackball, light pen, loudspeakers, microphone, etc.

[0148] Networking means 306 are selected from a device providing network data receiving and transfer, e.g. Ethernet-card, WLAN/Wi-Fi module, Bluetooth module, BLE module, NFC module, IrDa, RFID module, GSM modem, etc. Making use of the means 305 provides an arrangement of data exchange through wire or wireless data communication channel, e.g. WAN, PAN, LAN, Intranet, Internet, WLAN, WMAN or GSM.

[0149] The components of the device 300 are interconnected by the common data bus (310).

[0150] The application materials have represented the preferred embodiment of the claimed technical solution, which shall not be used as limiting the other particular embodiments, which are not beyond the claimed scope of protection and are obvious to persons skilled in the art.

[0151] Modifications and improvements of the above embodiments of this technical solution will be obvious to those skilled in the art. The foregoing specification is provided by way of example only and is not intended to be limiting in any way. Thus, the scope of this technical solution is limited only by the scope of the attached claims.

CLAIMS

1. A computer-implementable method of detecting compromised network devices based on DNS tunneling, the method comprising:

receiving a traffic of a protected network infrastructure;

extracting at least one DNS query from the traffic;

filtering a list of extracted DNS queries by excluding from the list a queries to a predefined domain name types;

feeding one-by-one each of a plurality of domain names remaining in the list as an input to a pre-trained decision rule;

if at least one domain name obtains a negative assessment of the decision rule, increasing a penalty for at least one network device sent at least one DNS query, containing this domain name;

if the amount of penalties of at least one network device exceeds a preset threshold, this at least one network device is considered to be compromised,

and displaying a corresponding notification.

2. The method of claim 1, wherein receiving the traffic of the protected network infrastructure also comprises receiving an inbound network traffic of the protected network infrastructure.

3. The method of claim 1, wherein the traffic is received by a traffic mirroring method.

4. The method of claim 1, wherein for each DNS query extracted from the traffic at least following data are extracted: a domain name; a sender IP address and a recipient IP address.

5. The method of claim 4, further comprises checking whether at least one recipient IP address is in a blacklist and in a whitelist, and

in response to at least one recipient IP address is in the blacklist, increasing the penalty for at least one network device which sent at least one DNS query, containing this at least one recipient IP address;

in response to at least one recipient IP address is in the whitelist, skipping the DNS query.

6. The method of claim 1, wherein the predetermined types of domain names include at least:

second level domain names,

low entropy domain names, and

knowingly trusted domain names from well-known reliable sources.

7. The method of claim 1, wherein the pre-trained decision rule comprises at least one the following heuristic methods: dividing domains by entropy level, selecting n-grams in domain names.

8. The method of claim 1, wherein the pre-trained decision rule is a LSTM neural network.

9. The method of claim 8, wherein LSTM neural network is additionally fed with statistics of a network interactions of the network device which sent the DNS

query, containing the given domain name, as well as the penalty of this network device,

wherein statistics of network interactions is collected during last T minutes.

10. The method of claim 1, wherein pre-trained decision rule is a combination of LSTM neural network and at least one of the following heuristic methods: dividing domains by entropy level, selecting n-grams in domain names.

11. The method of claim 1, further comprising:

searching for the at least one domain name obtained the negative assessment in the history of all inbound and outbound traffic from at least one network device of the protected infrastructure, and

in response to detection entries of the at least one domain name obtained the negative assessment:

either additionally penalizing every network device of the protected infrastructure which communicated with the given domain name,

or penalizing every network device which is not penalized before.

12. The method of claim 1, further comprising:

blocking the domain name obtained the negative assessment for every network device included into the protected infrastructure,

generating notifications on maliciousness of the blocked domain name,

creating template emails for notifying interested parties about blocked domain name,

adding the blocked domain name to at least one blacklist.

13. A computing device for detecting compromised network devices based on DNS tunneling, the computing device is configured to perform the method of detecting compromised network devices based on DNS tunneling according to any of claims 1-12.