

METHOD AND DEVICE FOR CLUSTERING PHISHING WEB RESOURCES BASED ON VISUAL CONTENT IMAGE

FIELD

[0001] This technical solution relates to the field of computing, in particular to the field of information security, namely, to a method and a computing device for clustering phishing web resources based on visual content image for further attribution of phishing websites, mirrors, blocked and/or illegal websites.

BACKGROUND

[0002] Phishing sites are one of the most widespread types of malicious web resources. Their main purpose is obtaining confidential user data to commit fraudulent acts.

[0003] The principle of phishing sites operation is copying or imitation of existing services of any brands, for example, banks, payment systems, e-mail, social networks or trademarks to obtain confidential data voluntarily from a user (login, password, e-mail, name, ID document number, payment card details, etc.).

[0004] For example, but not limited to, this could be a phishing remote banking site where a user is encouraged by various psychological tricks to enter login data and/or provide additional confidential information. After that, the obtained information is used by phishers for stealing user funds or other illegal actions.

[0005] It is worth noting that imitation of real services could also be carried out with the purpose of making money on advertising, displaying untrustworthy or banned information, as well as in case of an official resource mirror – for illegal distribution of copyrighted or banned content. All described actions may damage business reputation of the original web resource owners or provide illegal access to unsuitable content.

[0006] Phishing distribution techniques allow to create many same type sites using so-called phishing kits, which complicates their timely detection and blocking.

[0007] A phishing kit is an archive file comprising scripts required for creation and operation of a phishing site. Such a tool allows phishers without deep programming skills to deploy hundreds of phishing pages quickly, often using them as "mirrors" of each other. When such site is blocked, the phisher activates another one, when this one is blocked, the next one, and so on. Thus, a phishing kit allows attackers to resume operation of malicious resources quickly, ensuring their own invulnerability. This explains the interest in them among cyber security experts. Detecting phishing kits allows not only to find hundreds or even thousands of phishing pages, but what is more important, it could serve as a starting point for investigations in order to identify their developers and bring them to justice.

[0008] It is worth noting that in order to attract buyers, the developers of phishing kits use well-known brands with a large audience, which, in theory, should make it easier for future owners of such kits to implement fraudulent schemes. In 2020, the brands most frequently used in phishing kits were Amazon, Google, Instagram, Office 365, and PayPal, while Exploit, OGUsers and Crimenetwork were in top 3 online phishing kits markets.

[0009] At its simplest, phishing web resources are identified by comparing URL to databases of compromised URLs. More sophisticated methods could be based on the analysis of the page structural elements.

[0010] US patent No. 8856937 (publ. 07.10.2014) discloses a method and a system for identifying malicious web resources. This known method includes creating and updating a web resource database comprising legitimate web resources, suspicious web resources and malicious web resources. Web resources are classified based on comparing web resource screenshots to legitimate and phishing web resources contained in the database.

[0011] CN 101826105 B (publ. 02/05/2019) discloses a method for detecting phishing pages based on matching web page elements, namely, calculating the

optimal correspondence of graphs to search matching pairs of functions among various signatures of web pages. The method is also characterized by determining internal weights of text features, image characteristics by determining relative weights among text similarity, and images when calculating web pages similarity using logarithmic regression analysis.

[0012] In order to speed up the process of identifying malicious web resources, which could be performed using, for example, the systems and methods disclosed in the above prior art documents, web resource clusters could be formed comprising a selection of web resources which further are analyzed to identify, for example, web resources involved in phishing, among them.

[0013] Patents US 10200381 B2 (publ. 02/05/2019) and US 8381292 B1 (publ. 02/19/2013) disclose the methods of protecting a brand from phishing by comparing the profiles of a potentially phishing resource to the basic (required) templates of brand web pages.

[0014] One of the disadvantages of the known devices and methods for clustering web resources, for example, disclosed in the source US 2015/0067839 (publ. 03/05/2015), consists in excessive number of analyzed suspicious web resources contained in the obtained clusters, which in turn causes excessive labor intensity and/or resource consumption for checking all these suspicious web resources in the obtained clusters to identify among them the web resources that are copies of the original web resources or comprising at least some elements of the original web resource content, with their subsequent analysis for involvement in phishing activities.

[0015] Preparation of target web resource clusters, in which the web resources are highly likely to be copies of an original web resource and are involved, for example, in phishing activities, is an important aspect for improving speed and accuracy of identifying malicious web resources. Many known technical solutions are designed for the maximum accurate preparation of target clusters of web resources, however, as noted above, these known technical solutions have some disadvantages and require improvement.

[0016] The prior art also discloses the solution RU 2676247 C1 (publ. 12/26/2018), selected as a prototype, which solves the problem of automated filtering of an array of suspicious web resources to identify essentially all phishing web resources duplicating at least part of the content from a specific original web resource owing to an ability of creating a database comprising only the assumed phishing web resources, each of which could be further automatically and/or manually analyzed for being involved in malicious phishing-related activities. Automated filtration of the entire array of the identified suspicious web resources, in turn, enables to reduce significantly the computing resources of a computer device and its memory resources required for identifying involvement of the analyzed web resources in malicious actions related to phishing, due to the fact that not the entire array of suspicious web resources is subject to analysis, but only a limited set of web resources associated with the original web resource and highly likely to be related to phishing due to duplicating content of the original web resource being legitimate.

[0017] However, the said solution is aimed at clustering based on the elements retrieved during collection from the page, in contrast to this solution, where the elements are retrieved from the web resource image, that enables to improve accuracy of approach, since different phishing pages using different brands could have the same script (JavaScript) or the same icon, or the same signatures in the HTML code. The advantage is that already rendered content is subject to analysis, that is, the content obtained after the browser has performed certain steps to convert HTML, CSS and JavaScript into a visual image, but not the code data. This approach results in decreasing the frequency of false responses.

[0018] Thus, in future, it is necessary to improve devices and methods for clustering web resources, in particular for preparation of such web resource clusters that would reduce labor intensity and/or resource consumption of their subsequent checking for involvement, for example, in phishing, by using advanced computer vision technology in combination with an advanced clustering algorithm.

[0019] Therefore, this technical solution is created to overcome at least one of the above-mentioned disadvantages of the known devices and methods for clustering content of phishing web resources.

SUMMARY

[0020] The technical problem to be solved by the claimed solution is creation of a method and computing device for clustering phishing web resources based on visual content image. Additional variants of this technique implementation are presented in the dependent claims.

[0021] The technical result consists in improving accuracy of clustering phishing web resources based on visual content image.

[0022] The claimed result is achieved by implementing a method for clustering phishing web resources based on visual content image, executed on a computer device comprising at least a processor and memory, and the method comprises the steps of:

- receiving references to a set of phishing web resources;

- retrieving at least one image of the visual content of each web resource of the set;

- processing the content of each visual content image associated with one of the set web resources, while

- contouring the elements on each image of the phishing web resource visual content;

- filtering the identified contours in each visual content image by removing the identical contours;

- combining the web resource associated with the compared contours and the cluster based on pairwise comparison of the identified contours and cluster contours, wherein, if the similarity value overrides the threshold value,

- otherwise, creating a new cluster for the web resource;

- storing references to web resources associated with corresponding contours of the content from a set of specified clusters in a database.

[0023] In the specific embodiment of the described method, a set of web resources includes at least two web resources.

[0024] In the specific embodiment of the described method a network is preliminary scanned for obtaining references to the set of web resources.

[0025] In the specific embodiment of the described method the references to the set of web resources are obtained, at least partially, from an external source via a data transmission network.

[0026] In the specific embodiment of the described method the references to the set of web resources are obtained from the preliminary-formed web resource database.

[0027] In the specific embodiment of the described method there are at least the following contours of elements contained in the visual content image: logos, letter outlines, individual words, pictures, forms.

[0028] In the specific embodiment of the described method at least the following methods are used for contour extraction: Canny edge detector and contour tracing algorithm.

[0029] In the specific embodiment of the described method when processing the content of each obtained visual content image, small-size, oblique and elongated contours are additionally removed.

[0030] In the specific embodiment of the described method the identified contours are filtered in each visual content image by removing similar contours, and if the compared contours differ in height and/or width by more than two pixels, they are considered to be different

[0031] In the specific embodiment of the described method the identified contours are filtered in each visual content image by removing similar contours, and if the compared contours differ in height and/or width by two pixels or less, then

the contours are overlapped, and

calculating per-pixel difference of the images,

wherein, a similarity score is calculated for the resultant differential image within the range from 0, where the images are the same, to 100, where the images are completely opposite,

wherein, the similarity score (P) is calculated as the arithmetic mean of pixel values, that is, the sum of pixel values divided by the number of image pixels, and calculated for the values taken from all three RGB channels:

$$P = \frac{\sum_{i=1}^N Ri + \sum_{i=1}^N Gi + \sum_{i=1}^N Bi}{N},$$

where Ri, Gi, Bi – i-th pixel value in R, G and B channels, respectively, and N – total number of image pixels,

wherein, the contours which similarity score does not exceed 7 are removed.

[0032] In the specific embodiment of the described method, invariant visual hash functions are additionally calculated for each contour identified in the visual content image

[0033] In the specific embodiment of the described method, if, when comparing with the cluster images, less than three similar contours are found in the image of the phishing web resource visual content, then an undescriptive image message is displayed, and in this case, a new visual content image is not added to the cluster, and if there are less than two images in the cluster, the cluster is deleted;

if the comparison reveals from three to five similar contours, then, in order to add an image of the phishing web resource visual content to a specific cluster, all the cluster contours should match exactly;

if the comparison reveals from five to ten similar contours, then, in order to add an image of the visual content of a phishing web resource to a specific cluster, half of the contours must match with the cluster contours,

if the comparison reveals from ten to fifteen similar contours, then, in order to add an image of the phishing web resource visual content to a specific cluster, a quarter of the cluster contours should match,

if the comparison reveals minimum fifteen similar contours, then, in order to add an image of the phishing web resource visual content to a specific cluster, at least one fifth of the cluster contours should match.

[0034] In the specific embodiment of the described method, when attributing an image of the phishing web resource visual content to several clusters, it is added to a cluster with the largest number of contour matches.

[0035] In the specific embodiment of the described method after clustering each of the images of the phishing web resource visual content set, the clusters containing less than three images are additionally deleted.

[0036] In the specific embodiment of the described method the images contained in the remote clusters are entered into a special list and are temporarily considered as garbage.

[0037] In the specific embodiment of the described method an adjacency matrix is additionally created based on each image of the visual content contained in the list, where the number of matching contours is located at the intersection, in order to identify a list of pairs of contours similar in size.

[0038] In the specific embodiment of the described method, additionally, to determine the association of at least two images of web resource visual content in the adjacency matrix, the value is changed to 0, if the number of size-matching contours is less than a predetermined threshold, or to 1, if the number of contour matches is more than a threshold.

[0039] In the specific embodiment of the described method the strongly connected graph components, in which there are more than two images, are computed, and a new cluster is created for each of the identified components.

[0040] The claimed result is also achieved by implementing a computing device for clustering phishing content, comprising a memory for storing machine-readable instructions and at least one computing processor configured to execute machine-readable instructions to ensure implementation of the method for clustering phishing content.

DESCRIPTION OF THE DRAWINGS

[0041] 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:

[0042] Fig. 1 illustrates a simplified schematic diagram of one of non-limiting embodiments of the system implementing the present technical solution.

[0043] Fig. 2 illustrates one of non-limiting embodiments for representing information of web page resources in a resource database.

[0044] Fig. 3 illustrates a flow diagram according to the aspect of the present method for clustering phishing web resources based on visual content image.

[0045] Fig. 4 illustrates a non-limiting example of the first web resource.

[0046] Fig. 5 illustrates a non-limiting example of the second web resource.

[0047] Fig. 6 illustrates a non-limiting example of the third web resource.

[0048] Fig. 7 illustrates a simplified schematic diagram of one of non-limiting embodiments of a computing device for implementation of this technical solution.

DETAILED DESCRIPTION

[0049] Numerous implementation details intended to ensure clear understanding of this technique are given in the detailed technique 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 detail so as not to obscure unnecessarily the present technique.

[0050] 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.

[0051] Note that, in the context of the present specification, unless specifically stated otherwise, the term "content element" or "content" refers to any

data that could be represented (visually, in audio format, or in some other way) that a web resource could comprise. Thus, for the purposes of this technical solution, a content element could be, in particular, a font, menu item, web resource template, fill-out form, written text, image, part of the image, graphics, animation, video, music, voice record, and so on, and also any combination thereof. In particular, any other resources that can be transferred via http/https protocols are also content.

[0052] Additionally, the present specification discloses the term image of the web resource visual content, which means a screenshot, that is, an image obtained by a computing device and showing exactly what the user sees on the screen of a monitor or other visual output device.

[0053] Contours (boxes) - rectangular elements, for example, buttons, input fields, company logos, title blocks, etc. highlighted in the visual content image, that is, in the screenshot.

[0054] Clusters are groups of visual content images (screenshots), formed on the basis of matching the unique contours (boxes) identified on them, for a specific phishing set. A set of links (references) to phishing pages corresponding to the visual content images and having much in common (for example, directed to one specific company or similar to each other in landing), is also associated with each cluster, for example, stored in a database with a reference to a specific cluster.

[0055] In the context of the present specification, unless clearly specified otherwise, an "locator" or "reference" to an information element could be the information element itself or locator, reference, link or other indirect method enabling the recipient of the reference to find a network, memory, database or other computer-readable medium from which the information element can be retrieved. For example, a web resource locator could include the file itself or a set of web resource files, or it could be a universal resource locator (for example, URL such as www.webpage.com) identifying a web resource with respect to a particular network (in particular, the Internet), or it could by some other means transmit to the recipient a reference to a network folder, memory address, table in a database or other place where one can access a web resource or its individual content elements. Those

skilled in the art would appreciate that the accuracy degree required for such reference depends on the degree of initial understanding of how the information exchanged between the reference recipient and sender should be interpreted. For example, if prior to data transfer between the sender and the recipient, it is clear that the information element locator takes the form of a uniform resource locator URL, transfer of the locator, link to a given web resource – is all that is required for efficient transfer of the web resource to the recipient, despite the fact that the information element itself (for example, a web resource or its individual content element) has not been transferred between the reference sender and the recipient.

[0056] In the context of the present specification, unless clearly specified otherwise, the words the "first", the "second", the "third", etc. are used as adjectives solely to distinguish the nouns which they refer to from each other, and not for the purposes of describing any specific relationship between these nouns. Thus, for example, it should be kept in mind that the use of the terms the "first web resource" and the "third web resource" does not imply any order, assignment to a particular type, history, hierarchy or ranking (for example) of web resources in the web resource set, as well as their use (in itself) does not imply that some "second web resource" must necessarily exist in a particular situation. Hereinafter, as specified here in other contexts, the reference to the "first" element and the "second" element does not exclude a possibility that it is one and the same actual real element. Thus, for example, in some cases, the "first" server and the "second" server may be one and the same software and/or hardware, and in other cases they may be different software and/or hardware.

[0057] Additional and/or alternative characteristics, aspects and advantages of embodiments of this technical solution will become obvious from the following specification, attached drawings and claims.

[0058] Fig. 1 illustrates a simplified schematic diagram of one of non-limiting embodiments of the phishing content clustering system 100 for implementing this technical solution.

[0059] In some non-limiting embodiments of the present technical solution, the system 100 implementing the method may include a data transmission network 110, at least one computing device 120 for detecting malicious web resources, and a database 130.

[0060] The system 100 may also include at least one search server 140 that has access to the web resource database 130.

[0061] In the other non-limiting embodiment of the present technique, the system 100 may additionally include active network equipment (not shown) and a client device (not shown).

[0062] The Internet or any other computer network could be used as the data transmission network 110.

[0063] The computing device 120 could be a conventional computer server. In an exemplary embodiment of this technical solution, the device 120 could be Dell™ PowerEdge™ server, or it could be any other suitable hardware and/or application software and/or system software, or a combination thereof.

[0064] Possible embodiments of the computing device 120 are well known. Thus, it is sufficient to note that the device 120 includes, but not limited to, a network communication interface (e.g. modem, network card, etc.) for two-way communication over a data transmission network (not shown); and the processor 121 connected to the network communication interface, with the processor 121 being configured to execute various algorithms, including those described below. For this purpose, the processor 121 could have access to the computer-readable instructions stored in a computer-readable medium (not shown), the execution of which causes the processor 121 to implement various procedures, described hereinafter.

[0065] In some embodiments of this technical solution, the computing device 120 includes at least one database 122 that is configured to store at least part of references to content elements, content elements, group of content elements, and a reference to web resources associated with the corresponding content element or group of content elements. Thus, by way of example, a web resource that contains

all the content elements from the group of content elements is associated with that group of content elements.

[0066] It is worth noting that the database 122 could be implemented as part of the computing device 120 or otherwise be accessible for the computing device 120.

[0067] An embodiment of the system 100 is possible, according to which the computing device 120 includes a computer-readable medium (not shown) where computer-readable instructions and/or database 122 could be preliminarily stored. Alternatively or additionally, the device 120 could have access to the computer readable instructions and/or database 122 located in the other equipment. The way in which the processor 121 of the computing device 120 accesses the computer readable instructions and the database 122 is not a limiting condition of this technical solution.

[0068] The processor 121 is configured to receive references to a set of web resources. By way of non-limiting example, receiving references to a set of web resources could be receiving a list of uniform resource locators (URLs). Additionally, it is worth noting that the sources of references to a set of web resources could be different.

[0069] An embodiment of this technical solution is possible, in which the processor 121 is configured to receive a reference to a set of web resources at least in part from an external source over the data transmission network 110. For example, the external source could be a search server 140 accessing the web resource database 130 and/or malicious web resource data collection server (not shown), etc. Moreover, the computing device 120 could access the web resource database 130 over the data transmission network 110. Alternatively or additionally, the external source could be a web resource that includes references to a set of web resources (for example, a verified database of phishing web resources like Google Safe Browsing (GSB), OpenPhish (OP)). It is worth noting that the way in which and by what principle the references to web resources are grouped in the external source, in particular, in the web resource database 130, is not a limiting condition.

[0070] Embodiments of the search server 140 are well known. Thus, it is sufficient to note that the search server 140 includes, but not limited to, a network communication interface (e.g., a modem, a network card, and the like) for two-way communication over a data transmission network 110; and a processor (not shown) connected to the network communication interface, with the processor being configured to search for web resources in the data transmission network 110 and to store them in the database 130. For this purpose, the processor of the search server could have access to corresponding computer-readable instructions stored in a computer-readable medium (not shown).

[0071] Thus, the process of population and storing the web resource database 130 is generally known as data collection (or crawling), wherein the search server 140 is configured to browse various web resources available over the data transmission network 110, and store them in the web resources database 130 according to one or more predefined parameters. By way of non-limiting example, the search server 140 may scan the data transmission network 110 and store any new and updated web resources, and thus collect the complete web resource base 130 of the data transmission network, and/or collect web resources, for example, including certain keywords in one or more languages. Since malicious web resources (in particular, phishing websites) usually try to imitate the websites of banks, payment systems and e-mail services, the keywords could be, for example, but not limited to: "bank", "credit", "card", etc. The search could also be carried out by domain names and/or other parameters. Identified and selected references to web resources are stored in the web resource database 130, after that they could be received by the processor of the computing device 120 over the data transmission network 110 from the search server 140.

[0072] Moreover, the search server 140 is configured to explore at least part of the content elements of each web resource from a set of web resources. Web resource content elements could be files of content elements and/or hash sums of content element files (e.g. hash sums computed by sha256 algorithm). Format of content element files is not specifically limited and will depend on the type of

specific content element. For example, in case, where the content element is an image, the file could be presented, in particular, in one of the following formats: "jpg", "jpeg", "png", "bmp", "gif", etc. In case, where the content element is a text, the file could be presented, in particular, in one of the following formats: "txt", "doc", "html", etc. In case, where the content element is a font, the file could be presented, in particular, in one of the following formats: "woff", "ttf," eot "," svg ", etc. In case, where the content element is a script, the file could be presented, in particular, in one of the following formats: "asp", "aspx", "php", "jsp", "cgi", etc. In case, where the content element is animation or video, the file could be presented, in particular, in one of the following formats: "flv", "swf", "avi", "mp4", "mov", etc. In case, where the content element is a cascading style sheet, the file could be presented, in particular, in "css" format. The specified list of possible content elements and formats in which they could be presented is not a limiting condition of this technical solution.

[0073] The extraction of elements, and also creation of visual content images (screenshots) of the identified web resources could be executed by the processor of the search server 140 using specially written scenarios and/or scripts for browsers.

[0074] Besides, it is worth noting that in the other non-limiting embodiment of this technique, the described function of creating visual content images could be performed by the computing device 120.

[0075] In one of the specific embodiments of this technique, the visual content image (screenshot) could be obtained by using Selenium WebDriver, a tool for automating the actions of web browser using `< .save_screenshot (filename)>` or `<.get_screenshot_as_file (filename)>` method and be stored in the database 122.

[0076] Alternatively, an embodiment of this technical solution is possible, in which the processor 121 of the computing device 120 is configured to receive a reference to a set of web resources directly from the preformed web resource database 130. Moreover, the preformed web resource database 130 could be accessible for the computing device 120 over the network.

[0077] An embodiment of this technical solution is possible, in which the processor 121 is configured to scan the network 110 for receiving references to a set of web resources, and thus, the computing device 120 can perform the functions of the search server 140 described above. For example, network scanning could be performed by IP addresses of web resources, domain names, keywords and other parameters.

[0078] The processor 121 of the computing device 120 is configured to process the obtained elements and also images of the web resource visual content for clustering phishing content, and also to group them based on the specific similarity threshold values.

[0079] Fig. 2 shows an illustrative example of the database 122 comprising data on contours of content elements (122a) and web resources (122b) associated with the corresponding contours of content elements.

[0080] In the illustrative example of the database 122 there is the cluster of associated web resources 300, 400, 500 based on the identified element boxes 302, 303, 306, 307, 308, 311, 312, 313.

[0081] It should be noted that the quantity and quality of the database 122 population is not limited in any way, the given examples of content element boxes and associated web resources are only for purposes of understanding this technical solution and are not limiting. It is obvious, that populating the database 122 could be much more complex.

[0082] Then, with reference to Fig. 3, the claimed method for clustering phishing web resources based on visual content image, which is performed in accordance with non-limiting embodiments of this technical solution (200), will be considered in more detail.

[0083] The method 200 may be performed in the computing device 120, and more specifically, by its processor 121, in accordance with a non-limiting embodiment of the system 100 illustrated in Fig. 1. Also with reference to Fig. 4, Fig. 5 and Fig. 6, let us consider an illustrative embodiment of the method.

[0084] Fig. 4 illustrates an example of a virtual content image in respect to the web resource 300 including a set of dedicated contours 301-314. At the same time, the contours 302-306 represent a button element for redirecting from this resource to some third-party resource, the contours of the elements 301, 307-311 represent brand logos, the contours 312-313 – characters, and 314 – dedicated rectangular area comprising the rest of the elements.

[0085] Fig. 5 illustrates an example of a virtual content image of the second web resource 400 including a set of content elements. Moreover, some contours of the content elements, namely 302-314, are also present in the first web resource 300 and are described above. Additionally, the web resource 400 comprises elements 401, 402, 403.

[0086] Fig. 6 illustrates an example of the third web resource 500, including content elements, in particular, 301, 302, 303, 306, 307, 308, 311, 312, 313, 501. Content elements 302, 303, 306, 307, 308, 311 312, 313, which are also present in the first and second web resource, and also 301 associated only with the first web resource, have been described above. Content element 501 is a string comprising buttons to navigate to third-party web resources.

[0087] In the context of the present technical solution, the number and kinds of different content elements is not a limiting condition. The web resources in Fig. 4, Fig. 5 and Fig. 6 are presented only as illustrative examples, it is obvious that this technical solution can also be applied to any other web resources that have more or less content elements, wherein, the types of content elements and the content itself may or may not be repeated within one and the same web resource. Some content elements available in the presented illustrative examples of web resources are not numbered.

[0088] Fig. 3 illustrates a flowchart in accordance with an aspect of the present method (200) for clustering phishing web resources based on visual content image, which will be disclosed in more detail below.

STEP 210 RECEIVING REFERENCES TO A SET OF FISHING WEB RESOURCES

[0089] Accordingly, the method 200 begins at step 210 where the computing device 120 receives references to at least one web resource comprising phishing content.

[0090] References to a set of web resources could be as the web resources themselves, as, for example, links to web resources, URLs, accessible over the data communication network 110 or another network (not shown) for the computing device 120.

[0091] In a non-limiting illustrative example the computing device 120 at step 210 receives references to a set of three web resources 300, 400 and 500 from an external source – search server 140 from the web resource database 130. It is worth noting that the web resources 300, 400 and 500 were previously found by the search server 140 and stored in the web resource database 130. Also, references to a set of web resources could be obtained at least in part from the preformed web resource database 130, and additional embodiment of the method 200 is possible, in which the computing device 120 can receive references to a set of web resources directly from the web resource database 130.

[0092] Additionally, the other embodiment of the method 200 is possible, in which the computing device 120 scans the data communication network 110 to obtain references to a set of web resources. Scanning of the data communication network 110 could be performed by the computing device 120 similarly with the search server 140, as described above, or otherwise using available software capable of searching according to the specified parameters in the network.

[0093] The other embodiment of the method 200 is possible, in which at step 210 a set of web resources is obtained from at least two different sources. For example, references to web resources 300 and 500 are obtained from the search server 140 from the database 130, and web resource 400 is obtained from the other external source (not shown, for example, from a special external database of phishing web resources) over data communication networks 110.

[0094] Additional embodiment of the method 200 is also possible, in which the obtained URL leads to a folder with files (for example, js, css, image, etc.), and

does not correspond to usual display of web pages, then the system 100 over the network 110 connects to an external service (not shown), for example, to 'scan-open-dir', which examines all the folders found at the specified address, and also returns the result back to the system 100, which, using specially written parsers, processes this information and distributes it in the database 122.

[0095] Then the method 200 proceeds to step 220.

STEP 220 RETRIEVING AT LEAST ONE IMAGE OF THE VISUAL CONTENT OF EACH WEB RESOURCE OF THE SET.

[0096] Step 220 - retrieving at least one image of the visual content of each web resource of the set.

[0097] In non-limiting embodiments of this technique, an image of the visual content could be rendered by the search server 140 and stored in the database 130, or rendered by the processor 121 of the computing device 120 and stored in the database 122 by any well known technique.

[0098] In particular, for an illustrative example, such techniques can include use of virtual browsers (eg, Selenium). Extension is installed in the virtual browser settings, cache is cleared, transition to a web resource is performed, then visual content images of this web resource are automatically saved in the specified location. Moreover, any available software, a script written in any programming language and capable of being interpreted by the system for obtaining the desired visual content image, could be used for the same purposes.

[0099] Then, after the step of retrieving the visual content image, the method proceeds to step 230.

STEP 230 PROCESSING THE CONTENT OF EACH RETRIEVED IMAGE OF THE VISUAL CONTENT ASSOCIATED WITH ONE OF WEB RESOURCES OF THE SET.

[0100] At this step, there is processing the content of each retrieved image of the visual content associated with one web resource of the set.

[0101] This step in a preferred embodiment of the technique described herein includes at least the following substeps:

contouring the elements on each image of the phishing web resource visual content;

filtering the identified contours in each visual content image by removing the identical contours.

[0102] At this step, there is contouring of at least the following elements of the visual content image: logos, letter outlines, individual words, pictures, forms. This could be done using Canny edge detector and contour tracing algorithm.

[0103] It is worth noting that contours in the visual content image are such curves in the image along which there is sharp change in brightness, colour or other parameters of the image. Therefore, the goals of image conversion into a set of curves are: extraction of the essential image characteristics and reducing the amount of information for subsequent analysis.

[0104] Accordingly, during contouring by Canny edge detector, at least the following steps are performed:

removing noise and unnecessary details from the visual content image by applying the Gaussian filter, computing the first derivatives (magnitudes and directions) of the pixel intensity function in horizontal and vertical directions by applying the Sobel operator,

selecting pixels that potentially belong to the edge using the non-maximum suppression procedure, wherein, the pixels which the vectors of directional derivatives correspond to, being local maxima, are considered potential candidates for belonging to the edge;

and performing double clipping, that is, highlighting "strong" and "weak" edges, where pixels, which intensity exceeds the maximum threshold, are considered as pixels belonging to the "strong" edges. It is assumed that pixels with intensity in the range from the minimum to the maximum threshold value belong to "weak" edges. Pixels, which intensity is below the minimum threshold, are discarded, excluded from further consideration. The resulting edges comprise pixels of all "strong" edges and those pixels of "weak" edges, whose neighborhood comprises at least one pixel of "strong" edges.

[0105] It is worth noting that further the contour tracing algorithm is additionally applied on the visual content image. In this technical solution use of a certain algorithm for tracing image edges is not a limiting factor, therefore, it is possible to use any known method. As an example, such methods can include, but not limited to, Kirsch operator, Robins operator, Marr-Hildreth and Harris algorithms, and also its modifications: Shea-Thomas, Harris-Laplace, etc.

[0106] After contouring, small, inclined and highly elongated contours are additionally cleaned off (deleted). It is worth noting as an example, but not limited to, that contours, which are less than 350 pixels, could be considered as small, and contours, which width or height is less than 10 pixels are considered as highly elongated. Contours, which edges are not parallel to sides of the visual content image, are considered as inclined.

[0107] The cleaning criteria could be preset by an expert or generated automatically based on statistic data about use of content elements in a set of web resources. For example, if a preset threshold value of the number of web resources associated with a given content element is exceeded, it could be considered as a standard content element. Threshold value can be set manually by operator, or be selected using various automated algorithms, including machine learning ones.

[0108] In a preferred embodiment of this technique, the processor 121 of the computing device 120 performs pair-wise comparison of similar contours, wherein:

if the compared contours differ in height and/or width by more than 2 pixels, they are considered to be different;

if the compared contours differ in height and/or width by 2 pixels or less, then the contours are superimposed on each other and the per-pixel difference of contour images is calculated, and a similarity score is calculated for the resultant image within the range from 0, where the images are the same, to 100, where the images are completely different.

wherein, the contours which similarity score does not exceed 7 are removed.

[0109] In accordance with the above, the similarity score (P) is calculated as the arithmetic mean of pixel values (1), that is, the sum of pixel values divided by

the number of image pixels, and calculated for the values taken from all three RGB channels:

$$P = \frac{\sum_{i=1}^N Ri + \sum_{i=1}^N Gi + \sum_{i=1}^N Bi}{N} \quad (1)$$

where R_i , G_i , B_i – i -th pixel value in R, G and B channels, respectively, and N – total number of image pixels.

[0110] It is worth noting that in one embodiment of this technique the scale-invariant visual hash functions are used for each contour identified in the visual content image. Scale-invariant hashing is useful when, for example, one and the same visual content is displayed at different scales on different device hardware and operating systems. These hash functions themselves could be any well-known functions.

[0111] In one of the alternative embodiments it is possible to apply the similar contour removal method, wherein:

if the compared contours differ in height or width by more than 2 pixels, they are considered to be different;

if one of the contours is larger than the other one, then, 9 all possible locations of the smaller contour in the larger one are successively compared, difference between the superimposed contours is calculated, the similarity score (1) is calculated for the difference, and the contours, which difference does not exceed the set threshold, are removed.

[0112] It is worth noting that when performing the alternative embodiment of this technique, there could be obtained less than 9 locations of one contour in the other one, depending on the difference in one of the coordinates.

[0113] In the alternative embodiment of this technical solution, in case when one of the contours is larger than the other one, the difference in width is taken as *delta*, and at least one location is searched to obtain a per-pixel difference of images with displacements: $0, \frac{\text{delta}}{2}, \text{delta}$ in relation to the upper left corner. The same actions could be performed in case when the contours differ in height.

[0114] In an illustrative embodiment of this technical solution, in the case where the first contour is larger in width and the second one in height, the contours are aligned in width using at least the following displacement options: $0, \frac{\delta}{2}, \delta$ in relation to the upper left corner and then they are compared, so that the first contour lies inside the other one. Accordingly, the maximum difference in height/width can be 2 pixels, therefore, displacements, for example, in 0, 1 and 2 pixels are considered, while the contour that is larger in width/height is clipped according to the corresponding size of the smaller contour, that is, if the difference between the contours was 2 pixels wide, then the large contour is clipped in three ways: 2 pixels on the right; by 1 pixel on the left and on the right; and 2 pixels on the left. Then, when comparing the obtained contours, the similarity score (1) is calculated, and if the contours are similar, one of them is removed. Moreover, in this case, if in any of 9 overlays the score is less than the threshold, then the contours are considered to be similar.

[0115] Then, the method proceeds to step 240.

STEP 240 PAIRWISE COMPARING THE IDENTIFIED CONTOURS AND CLUSTER CONTOURS, AND IN RESPONSE TO THE FACT THAT THE SIMILARITY SCORE HAS EXCEEDED THE THRESHOLD VALUE, COMBINING THE VISUAL CONTENT IMAGE, COMPRISING THE COMPARED CONTOURS, WITH THE CLUSTER, OTHERWISE, CREATING A NEW CLUSTER FOR THIS VISUAL CONTENT IMAGE.

[0116] At step 240 the identified contours and contours belonging to the clusters are pairwise compared, and if the similarity score (1) exceeds the threshold value, the visual content image is added to the cluster; if the score does not exceed the threshold value, then a new cluster is created for this visual content image.

[0117] In non-limiting embodiments of the present technique, the clusters are formed groups of images associated with web resources based on the similarity of the identified unique contours (boxes).

[0118] In non-limiting embodiments, the cluster structure is at least a collection of immediate images of the visual content associated with a reference to it, and also the contours identified in the image, while additionally for each contour in the cluster, time of its last match and the number of subsequent mismatches are stored when adding a new image to the cluster.

[0119] As an illustrative example, as mentioned previously, after adding the first image of the visual content 300 to the cluster, all the contours highlighted therein become the cluster contours. Then, when adding an image of the visual content 400, the contours 302, 303, ... 314 match with the contours identified in the image 300, and time of their match is assigned to them. Number 1, which indicates that this contour has not matched when adding a new image to the cluster once, is assigned to the contour 301, and the contours 401, 402, 403 are added to the cluster contours. Then, when adding an image of visual content 500 to this cluster, the contours 301, 302, 303, 306, 307, 308, 311, 312, 313 are assigned to new time of their match, and for the contours 304, 305, 309, 310, 314, 401, 402, 403 the mismatch counter is increased by one; with the contour 501 being added to the cluster contours.

[0120] In accordance with this technical solution, after pairwise sequential comparison of visual content images, in case of exceeding the threshold value, a contour is added to the cluster, and in the described case all new contours, corresponding to the image of the visual content associated with the cluster, are added to the database 122. In this case, exceedance of a preset threshold value by the similarity score $P(1)$ is considered to be a match.

[0121] By way of example, but not limited to, with reference to Fig. 3, 4, 5, the method 200 starts its operation by processing the first image of the visual content illustrated in Fig. 3, then, as mentioned above, the said image is processed, identifying contours on it. After that, the identified contours are pairwise compared with the cluster contours (at the described moment the cluster contours are absent in the database 122 and their counter is set to 0), and a new cluster is created comprising the visual content image illustrated in Fig. 3, since the similarity score is $P(1)$ in

none of the cases does not exceed the similarity threshold value. In this case, the created cluster comprises an image of visual content, a reference to it, and also all the contours identified in the image, namely 301, 302 ... 312. Then, the method proceeds to clustering the second visual content image illustrated in Fig. 4. It is worth noting that this visual content image has identical contours with the cluster formed earlier based on processing of the first visual content image (302 ... 313). Number of matching contours of the web resource 400 and contours of the cluster created on the basis of the visual content image of the web resource 300 exceeds the allowable threshold, which means that the image of the web resource 400 is added to the cluster, while the contours 401, 402 identified only for the web resource 400 are added to the cluster contours.

[0122] An embodiment of this technical solution is possible, in which the processor 121 is configured to process each contour in the multithreading mode, namely to find in the database 122 all contours, which difference in size does not exceed a preset number of pixels, for example, two pixels in width and/or height, and to form many similar contours from them.

[0123] Accordingly, in embodiments of the described technique at least two tables are formed in the database 122, wherein the first table includes information on the number of images in which each of the contours identified in the visual content images was detected, to each available cluster, and the second table shows the total number of contours included in each cluster. Based on the information contained in the formed tables, the clusters, in which the total number of matching contours is less than 85% of the number of visual content image contours associated with it, are deselected (removed). In this case, if after removal there are more than fifteen clusters subject to further verification, then, fifteen clusters with the largest number of matches are selected from them.

[0124] Exemplary variants of the above tables are given below, namely Table 1, which includes the information about belonging of each identified contour to each of the clusters, and Table 2, containing information about the total number of contours in each cluster. It is also worth noting that the specific data in Tables 1 and

2 are given for illustrative purposes only, instead of these data there could be any other numerical values obtained when performing the method 200, among other things, the tables may contain other numbers of rows and columns.

Table 1

	Cluster 1	Cluster 2	Cluster N
Contour 1	3	5	10
Contour 2	7	8	9
Contour 3	0	1	1

Table 2

	Cluster 1	Cluster 2	Cluster N
Total number of contours associated with the cluster	70	90	40

[0125] In an embodiment of the present technical solution, further, in the multithreading mode, the contours of each cluster stored in the database 122 and the contours of the analyzed visual content image are additionally compared, while combining the matched contours and highlighting all the clusters comprising the matched contours in accordance with Table 1.

[0126] Moreover, at this step, in order to add a visual content image to the cluster, each of the contours identified in the image is compared in multithreading mode with each of the contours present in different clusters.

[0127] In possible embodiments of this method, calculating a threshold for adding a visual content image to the cluster (or combining several clusters) is performed as follows:

finding (min) minimum number of contours in the combined objects, that is, the smallest of two numbers: the number corresponding to the number of contours in the visual content image, and the number corresponding to the number of contours in the given cluster,

if $\min \leq 2$, then the computing device 120 could output a warning message about the undescriptive image, and in this case, a new image of the visual content is not added to the cluster;

$2 < \min < 5$, then, in order to add a visual content image to a specific cluster, it is required that all the contours in the image and contours in the cluster are matched;

$5 \leq \min \leq 10$, then, in order to add a visual content image to a specific cluster, it is required that at least 50% of the min contours in the image and contours in the cluster are matched,

$10 < \min \leq 15$, then, in order to add a visual content image to a specific cluster, it is required that at least 25% of the min contours in the image and contours in the cluster are matched,

$\min > 15$, then, in order to add a visual content image to a specific cluster, it is required that at least 20% of the min contours in the image and contours in the cluster are matched.

[0128] It is worth noting that the cluster is created only from the visual content images in which at least three contours are identified.

[0129] In embodiments of the present solution, it is possible to assign a visual content image to several clusters, with adding it to the cluster with the highest number of contour matches. Besides, this embodiment provides for verifying a possibility of combining the identified clusters.

[0130] It is worth noting that in possible embodiments of this technical solution, the steps of clusters combining are similar to the steps of adding an image of the web resource visual content to a cluster, with the difference that the contours of two clusters are compared pairwise, and, if the threshold is exceeded, the clusters are combined. Accordingly, all the contours of the first cluster are compared with all the contours of the second one, while there could be only one match for each contour, that is, if the contour of the first cluster matches with the contour of the second cluster, these contours are not considered in further comparisons.

[0131] The method 200 proceeds to step 250.

STEP 250 STORING REFERENCES TO WEB RESOURCES, WHICH VISUAL CONTENT IMAGES WERE FOUND TO BELONG TO THE SAME CLUSTER, IN THE DATABASE.

[0132] Then, the references to web resources, which visual content images during the above analysis were found to belong to the same cluster, are stored in the database 122 of the computing device 120. In other words, at this step, the information about what namely web resources belong to one and the same subset of the presumably phishing web resources is stored in the database.

[0133] In the other embodiment, the clusters identified and stored in the database could then be transmitted to external devices (not shown in the figures) for further attribution of phishing web resources, web resource mirrors, blocked or illegal websites.

[0134] It is worth noting that in the additional embodiment of this technique, after clustering all the visual content images, the clusters comprising less than the specified number of images, for example, for better understanding, less than three images of visual content, are deleted. Such images are entered into a special list stored in the database 122, and are temporarily considered as "garbage", un-descriptive.

[0135] Then, at least once per certain period of time, set in the system 100, a list from 1 to N of all remaining "garbage" images of visual content and their corresponding contours is obtained for filtering un-descriptive images of web resources.

[0136] After filtering, an additional matrix is created for the remaining visual content images in the list from 1 to N in the database, in which the remaining "garbage" images from 1 to N are represented both in rows and columns, and number of matching contours is filled in at the intersection of rows and columns. The generated adjacency matrix is used for identifying a list of unique pairs of contours similar in size and associated with the images by the device 120.

[0137] Table 3 shows a non-limiting example of generation of an additional matrix as exemplified in Table 3. It is also worth noting that the specific data in

Table 3 are given for illustrative purposes only, instead of this data there could be any other numerical values obtained when performing the method 200, among other things, the matrix may contain other numbers of rows and columns.

Table 3

	Image 1	Image 2	Image N
Image 1	0	19	12
Image 2	19	0	5
Image N	12	5	0

[0138] In a non-limiting embodiment of the present technique, after the device 120 has identified a list of similar-sized contours, the contours comprised in the list are compared in multiple threads, and only the most similar pairs of contours are left. The information about contours intersection is entered into the matrix stored in the database 122 for those visual content images which these contours belong to.

[0139] In this case, for each unique pair of images contained in the list, for example, with reference to Table 3, for image 1 and image 2, the contours are superimposed on each other to identify the number of similar pairs of contours and, if in some contours pair the contours are similar to each other, then, in the initially generated matrix, where everything is filled with 0, the number in the corresponding cell is increased by 1. In this example, as can be seen from Table 3, 19 similar pairs of contours have been found for this pair of images.

[0140] Then, in order to confirm the association of web resource pairs based on pairs of their visual content images, the matrix is checked line by line, while transforming it into the adjacency matrix. If the number of contours matched in each pair of images exceeds a preset threshold, one is put down in the corresponding cell of the matrix. If the number of matched contours is less than a preset threshold, zero is put down in the corresponding cell. A non-limiting example of the transformed adjacency matrix, shown above in Table 3, and processed in this way at the threshold value of 10 is shown in Table 4.

Table 4

	Image 1	Image 2	Image N
Image 1	0	1	1
Image 2	1	0	0
Image N	1	0	0

[0141] In non-limiting embodiments of this technique the strongly connected components of the adjacency matrix are further calculated. In fact, at this step, a maximally connected subgraph is constructed, where any vertex is reachable from any other graph vertex. Actually, the methods for calculating the strongly connected components of the adjacency matrix are well known in graph theory; however, any of these methods can be used for implementation of this technique.

[0142] In a non-limiting embodiment of this technique, the Tarjan algorithm with modifications of Nuutil or non-recursive Tarjan algorithm is used to determine the strongly connected components.

[0143] Each such constructed subgraph forms a new cluster of visual content images. Additionally, it is worth noting that a cluster constructed from "garbage" images of visual content is added to previously constructed clusters as an equivalent one. In this case, visual content images remained in the list of "garbage" images remain there until they are entered into one of new clusters constructed of "garbage" images.

[0144] Fig. 7 hereafter illustrates a schematic diagram of the computer device (700) processing the data required for embodiment of the claimed solution.

[0145] In general, the device (700) comprises such components as: one or more processors (701), at least one memory (702), data storage means (703), input/output interfaces (704), input/output means (705), networking means (706).

[0146] The device processor (701) executes main computing operations, required for functioning of the device (700) or functionality of one or more of its

components. The processor (701) runs the required machine-readable commands, comprised in the random-access memory (702).

[0147] The memory (702), typically, is in the form of RAM and comprises the necessary program logic ensuring the required functionality.

[0148] The data storage means (703) could be in the form of HDD, SSD, RAID, networked storage, flash-memory, optical drives (CD, DVD, MD, Blue-Ray disks), etc.

[0149] Interfaces (704) 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 (704) depends on the specific device (700), which could be a personal computer, mainframe, server cluster, thin client, smartphone, laptop, etc.

[0150] As input/output means (705) there could be used: keyboard, joystick, display (touch-screen display), projector, touch pad, mouse, trackball, light pen, loudspeakers, microphone, etc.

[0151] Networking means (706) 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 (705) provides an arrangement of data exchange through wire or wireless data communication channel, e.g. WAN, PAN, LAN, Intranet, Internet, WLAN, WMAN or GSM.

[0152] The components of the device (700) are interconnected by the common data bus (710).

[0153] 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.

[0154] The claimed method for clustering phishing content improves accuracy of clustering a set of web resources into groups, that further allows an expert to perform runtime checking of each saved group and to identify groups of phishing

sites, for example, created with the use of phishing kits, identify mirrors and copies of blocked and previously identified malicious web resources, identify the authors of a phishing kit or the owner(s) of a phishing web resource. This significantly reduces the load on an expert when identifying malicious web resources.

[0155] 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 method for clustering phishing web resources based on visual content image, the method is executed on a computer device comprising at least a processor and a memory, the method comprises:

receiving references to multiple phishing web resources;

retrieving at least one image of the visual content of each phishing web resource from the multiple phishing web resources;

processing the each image associated with one phishing web resource from the multiple phishing web resources, including

identifying contours of elements in each image; and

filtering the identified contours in each image by removing all but one instance of the identical contours;

if a similarity value exceeds a threshold value: based on pairwise comparison of the identified contours and one or more contours of an existing cluster for a specific phishing set, combining in a database the phishing web resource associated with the identified contours and the existing cluster, otherwise, creating a new cluster in the database for the phishing web resource;

storing references to the phishing web resources in the database.

2. The method of claim 1, further comprising preliminary scanning a network for obtaining the stored references.

3. The method of claim 1, further comprising obtaining the stored references, at least partially, from an external source via a data transmission network.

4. The method of claim 1, further comprising obtaining the stored references from a preliminary-formed web resource database.

5. The method of claim 1, wherein the elements include: logos, letter outlines, individual words, pictures, and/or forms.

6. The method of claim 1, wherein the contours are identified using at least the following methods: Canny edge detector and/or contour tracing algorithm.

7. The method of claim 1, further comprising removing small-size, oblique and elongated contours when processing each image.

8. The method of claim 1, wherein filtering the identified contours in each image includes removing similar contours, and if any compared contours differ in height and/or width by more than two pixels, the contours are considered to be different.

9. The method of claim 1,

wherein filtering the identified contours in each image includes removing similar contours,

wherein, if the compared contours differ in height and/or width by two pixels or less, then the contours are overlapped, and the method further comprises calculating per-pixel difference of the images,

wherein, a similarity score for the resultant differential image is within a range from 0, where the images are the same, to 100, where the images are completely opposite,

wherein, the method further comprises calculating the similarity score (P) as the arithmetic mean of pixel values, that is, the sum of pixel values divided by the number of image pixels, and calculated for the values taken from all three RGB channels:

$$P = \frac{\sum_{i=1}^N Ri + \sum_{i=1}^N Gi + \sum_{i=1}^N Bi}{N},$$

where Ri, Gi, Bi – i-th pixel value in R, G and B channels, respectively, and N – total number of image pixels, and

wherein, the method further comprises removing all but one instance of the contours with a similarity score that does not exceed 7.

10. The method of claim 1, further comprising calculating invariant visual hash functions for each contour identified in the image.

11. The method of claim 1, further comprising displaying an image message if, when pairwise comparing the images, less than three similar contours are found,

and a new image is not added to the cluster, and if there are less than two images in the cluster, the cluster is deleted;

if the comparison reveals from three to five similar contours, then, in order to add an image to a specific cluster, all the cluster contours must match exactly;

if the comparison reveals from five to ten similar contours, then, in order to add an image to a specific cluster, at least half of the contours must match the cluster contours,

if the comparison reveals from ten to fifteen similar contours, then, in order to add an image to a specific cluster, at least a quarter of the contours must match the cluster contours,

if the comparison reveals fifteen or more similar contours, then, in order to add an image of the phishing web resource visual content to a specific cluster, at least one fifth of the contours must match the cluster contours.

12. The method of claim 1, wherein the phishing web resource is combined with the existing cluster with the largest number of contour matches.

13. The method of claim 1, wherein after clustering each of the images, the method further comprising deleting the clusters containing less than three images.

14. The method of claim 13, further comprising entering the images contained in the deleted clusters into a special list and are temporarily considered as garbage.

15. The method of claim 14, further comprising creating an adjacency matrix based on the number of matching contours for each image of the special list.

16. The method of claim 15, wherein, the number of matching contours in the adjacency matrix, is changed to 0, if the number of matching contours is less than a predetermined threshold, or to 1, if the number of matching contours is more than the threshold.

17. The method of claim 16, further comprising computing strongly connected graph components, in which there are more than two images, and creating a new cluster for each of the identified components.

18. The method of any preceding claim, further comprising transmitting the database to an external device.

19. The method of any preceding claim, further comprising storing the identified contours in the database.

20. A computing device for clustering phishing content, the device comprising a memory for storing machine-readable instructions and at least one computing processor configured to execute machine-readable instructions to ensure implementation of the method for clustering phishing content according to any of claims 1-19.