

METHOD FOR IDENTIFYING MALICIOUS WEB RESOURCES

TECHNICAL FIELD

These techniques encompass methods and systems for early warning about cyber threats, especially methods to detect malicious web resources, in particular phishing websites, and mirrors of blocked and/or prohibited websites.

BACKGROUND

Phishing is widespread among malicious web resources. It allows to obtain confidential user data in order to commit fraudulent actions.

The operating principle of phishing sites consists of copying or imitating sites for various services, for example, banks, payment systems, online auctions, email services, and social networks in order to obtain confidential data from users voluntarily (account name, password, email, full name, passport number, credit card data, etc.). A user transferred to a phishing site, for example, a fake online banking service, is prompted to enter registration data and/or provide additional confidential information using various psychological techniques. For example, user is prompted to enter their account name and password for their online banking account, and to inform of the approximate balance in the account. Then scammers to steal the user's funds or take other illegal actions use the obtained information.

Furthermore, it is possible to copy or imitate web resources for purposes of mischief, profiting of displaying advertisements, displaying inaccurate or prohibited information, etc., which may cause reputational harm to the owners of the original web resources that were copied or imitated. Also, in the case of creating mirrors of prohibited or blocked web resources, users may be provided access to such resources, which may cause harm to third parties. For example, the owners of intellectual property may be harmed by the provision of access to mirror sites that distribute pirated content that was previously blocked.

Phishing sites are widespread in the area of online payment systems and banking services. Access to the targeted resources may be obtained through the devices of company employees, partners, and clients. The creators of phishing websites use various means to redirect user traffic to fraudulent pages: distributing spam, capturing traffic by interacting with the DNS server, typosquatting, etc.

According to the study Wombat Security State of the Phish 2017, in 2016 76% of organizations had been victims of a phishing attack. The total number of known phishing attacks in 2017 is over 1 million. In recent years, phishing technologies have been actively developed, the complexity of algorithms has increased, and the masking of phishing web pages has improved. It has become more difficult to identify phishing web pages, even for experienced users and the owners of web resources themselves. Modern internet security systems can identify phishing web resources, block them, and/or warn users when they are visiting suspicious web resources to caution them against entering confidential data.

Phishing distribution technologies facilitate the creation of many similar site using so-called **phishing packs (Phishing pack, Phishing Toolkit)**, which makes it more difficult to timely detect and block them. A phishing kit is a program that simplifies the creation of phishing sites; it contains the basic templates and scripts that are necessary to create and configure multiple phishing sites and to collect user data. By adding content from the original site, or content similar thereto, scammers can quickly create many similar phishing sites, which makes it more difficult to timely detect and block them. Due to the large quantity of phishing websites and their similarity to original web resources, it has become time-consuming and laborious for IT security specialists to check suspicious web resources manually. Creating automated systems and methods that accelerate the process of identifying and blocking phishing web resources is an important task.

In the simplest case, phishing web resources are identified by comparing the URL address with a database of URL addresses. Methods that are more complicated may be based on analyzing the page content.

US Patent Application No. 20120158626 (published June 21, 2012) describes a method for identifying and classifying malicious web resources. The method is based on applying machine learning algorithms to recognize and classify malicious web resources based on various characteristics.

US Patent No. 8856937 (issued October 7th, 2014) describes a method and system for identifying malicious web resources. This known method includes creating an update for a database containing legitimate web resources, suspicious web resources, and malicious web resources. Web resources are classified by comparing screenshots of web resources with legitimate and phishing web resources contained in the database.

In practice, known methods to identify malicious web resources automatically may incorrectly classify a web resource, for example, a legitimate web resource or a web resource

associated with a legitimate website will be classified as malicious. And, vice versa, some malicious web resources may not be identified as malicious or suspicious, but may be classified as legitimate, as a result of which all similar malicious web resources will also be classified as legitimate, for example, those created by the same phishing pack may also be classified as legitimate, non-malicious web resources. To prevent such situations from arising, an expert must manually verify the web resources. Manually verifying many web resources is laborious and can lead to erroneous results due to the human factor.

The objective of the proposed techniques is to create a method and system to identify malicious web resources. The proposed techniques may aim to reduce human involvement in the process of identifying malicious web resources, in particular, phishing web pages, and to save them in a database for further analysis.

DISCLOSURE

The speed and accuracy of identifying malicious web resources are important characteristics, which many known technical solutions are aimed at improving. However, as noted above, the known technical solutions possess shortcomings and need to be improved.

The first object is a method to identify malicious web resources, executed on a computer, the method comprising:

- receiving pointers for a plurality of web resources;

- extracting at least part of the content elements of each web resource in the plurality of web resources;

- combining at least two content elements into at least one first subgroup in response to the quantity of web resources including said content elements exceeds a predefined minimum threshold of web resources for that subgroup;

- combining at least two content elements into at least one second subgroup in response to the quantity of web resources including said content elements exceeds a predefined minimum threshold of web resources for that subgroup;

- combining the first and second subgroups into a group of content elements in response to the quantity of web resources including the content elements of the first and second subgroups exceeds a predefined threshold of web resources for that subgroup;

saving the content element group and the pointers for the web resources associated with the relevant content element group in the database.

In one possible embodiment of the method, the plurality of web resources contains at least two web resources.

In another possible embodiment of the method, a preliminary scan of the network is performed in order to obtain the pointers for the plurality of web resources.

In another possible embodiment of the method, the pointers for the plurality of web resources are obtained at least partially from an external source via a data network.

In another possible embodiment of the method, the pointers for the plurality of web resources are obtained from a previously generated database of web resources.

In another possible embodiment of the method, at the step of receiving the pointers for the plurality of web resources, at least some are obtained from previously extracted pointers for content elements of the plurality of web resources.

In another possible embodiment of the method, the content elements are files of content elements or hash sums of files of content elements.

In another possible embodiment of the method, after extracting at least some of the content elements, a filtration is performed by deleting previously defined standard content elements, and then at least two elements are combined into at least one subgroup without taking into consideration the deleted content elements.

In another possible embodiment of the method, after saving the group of content elements and pointers for web resources associated with said group of content elements, malicious web resources associated with the group of content elements are also identified.

In another possible embodiment of the method, a human expert performs the identification of malicious web resources at least partially.

In another possible embodiment of the method, after saving of each group of content elements and pointers for web resources associated with said group of content elements in the database, pointers are obtained for at least one additional web resource; at least some of the content elements of each web resource is extracted; in response to an identification in the database of at least one group of content elements associated with at least one specified additional web resource, the pointers for the additional web resource are saved in the database.

The second object is a computer device that identifies malicious web resources, connected to a data network and at least one database, including a computer processor, where the processor is configured to:

- receive pointers for a plurality of web resources;

- extract at least some of the content elements of each web resource in the plurality of web resources;

- combine at least two content elements into at least one first subgroup in response to the quantity of web resources that include said content elements exceeds a predefined minimum threshold of web resources for that subgroup;

- combine at least two content elements into at least one second subgroup in response to the quantity of web resources that include said content elements exceeds a predefined minimum threshold of web resources for that subgroup;

- combine the first and second subgroups into a group of content elements in response to the quantity of web resources that include the content elements of the first and second subgroups exceeds a predefined threshold of web resources for that subgroup;

- save the content element group and the pointers for the web resources associated with the relevant content element group in the database.

In one possible embodiment, the processor is configured to perform a network scan to obtain pointers for a plurality of web resources.

In another possible embodiment, the processor is configured to obtain pointers for the plurality of web resources at least partially from an external source via a data network.

In another possible embodiment, the processor is configured to obtain pointers for the plurality of web resources from a previously generated database of web resources.

In another possible embodiment, the processor is configured to extract at least some of the content elements of the web resource, wherein the content elements are files of content elements or hash sums of files of content elements.

In another possible embodiment, in the step of receiving pointers for the plurality of web resources, the processor is configured to obtain at least part of previously extracted pointers for content elements of the plurality of web resources.

In another possible embodiment, after extracting at least at least part of the content elements, the processor is configured to perform a filtration by deleting previously defined standard content elements, and to combine at least two elements into at least one subgroup without taking into consideration the deleted content elements.

In another possible embodiment, the processor is also configured to identify malicious web resources associated with a group of content elements.

In the context of this description, "computer device" means a computer program operating on relevant hardware, which is able to receive requests (for example, from client devices) via a network and to execute these requests or initiate the execution of these requests. The equipment may consist of one physical computer or one physical computer system, but neither is required for this techniques. In the context of this description, the use of the expression "computer device" does not mean that each request (for example, received commands or requests) or any particular request will be received, executed, or have its execution initiated by the same "computer device" (that is, the same software and/or hardware); this means that any number of elements of software or hardware may be involved in receiving/transmitting, executing or initiating the execution of any request or consequence of any request related to the client device, and all this software and hardware may consist of a single computer device or several computer devices, both options are included in the expression "at least one computer device."

In the context of this description, "database" means any structured plurality of data, regardless of the specific structure, the software used to manage the database, the computer hardware on which the data is stored, used, or otherwise made accessible for use. The database may be located on the same hardware that runs the process that stores or uses the information stored in the database, or it may be located on separate hardware, for example, a dedicated server or plurality of servers.

In the context of this description "component" means software (corresponding to specific hardware context) that is necessary and sufficient to perform the specific specified function(s).

In the context of this description, unless expressly indicated otherwise, the term "content element" or "content" means any data that may be presented (visually, in audio format, or otherwise), that a web resource may contain. Thus, for purposes of this solution a content element may be, in particular, a script, font, menu element, web resource template, fillable form, written text, an image, part of an image, a graph, animation, video, music, voice recording, etc., as well as any combination of these. Furthermore, the definition also encompasses any other resources that may be transferred via http/https protocols.

In the context of this description, unless expressly indicated otherwise, the "locator" or "pointer" of an information element may be the information element itself or a locator, reference,

link, or other indirect method that allows the recipient of the pointer to find the network, memory, database, or other machine-readable medium from which the information element may be extracted. For example, a web resource locator may include a file itself, or a plurality of web resource files, or it may even be a uniform resource locator (for example, URL www.webpage.com) that identifies the web resource in relation to a specific network (in particular, the Internet), or some other means to transfer the pointer to the recipient for a network folder, memory address, table in a database, or other place in which the web resource or its individual content elements can be accessed. As will be understood by those skilled in the art, the degree of specificity required for such an pointer depends on the degree of originally understanding how the information exchanged by the sender and receiver of the locator should be interpreted. For example, if it is understood prior to transmitting data between the sender and recipient that the information element locator takes the form of a Universal Resource Locator (URL), transmitting the locator and link to this web resource is all that is necessary to effectively transmit the web resource to the recipient, despite the fact that the information element itself (for example, the web resource or a specific content element of it) was not transmitted between the sender and recipient of the pointer.

In the context of this description, unless expressly indicated otherwise, the words "first," "second," "third," etc. are used as adjectives solely in order to distinguish the nouns they modify from one another, and not for purposes of describing any specific relation between these nouns. So, for example, it should be borne in mind that the use of the terms "first web resource" and "third web resource" do not imply any sort of order, assignment to a specific category, chronology, hierarchy, or ranking (for example) of web resources within the plurality of web resources, similarly their use (by itself) does not imply that a certain "second web resource" must exist in any given situation. Furthermore, as specified here in other contexts, mentioning a "first" element and a "second" element does not rule out the fact that they could actually be the same element. Therefore, for example, in some cases, the "first" server and the "second" server may be the same software and/or hardware, and in other cases, they may be different software and/or hardware.

In the context of this description "machine-readable medium" means a medium of absolutely any type and nature, including RAM, ROM, disks (CDs, DVDs, floppy disks, hard disks, etc.), USB flash drives, solid state drives, tape drives, etc.

Additional and/or alternative features, aspects, and advantages of different embodiments of this techniques will be apparent from the following description, drawings, and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to better understand these techniques, as well as its other aspects and characteristic features, reference is made to the following description, which should be used in conjunction with the accompanying drawings, where:

Fig. 1 depicts a system suitable for implementing non-limiting embodiments.

Fig. 2 shows one of the non-limiting implementations of displaying information about the resources of web pages in the resource database.

Fig. 3 is a flowchart that illustrates the aspect of this method for identifying malicious web resources.

Fig. 4 is a non-limiting example of a first web resource.

Fig. 5 is a non-limiting example of a second web resource.

Fig. 6 is a non-limiting example of a third web resource.

Fig. 7 is a non-limiting example of a fourth web resource.

DETAILED DESCRIPTION

The following description serves only a description of an illustrative example of these techniques. This description is not intended to define the scope or impose limitations on these techniques.

Some useful examples of modifications to the described method and system for identifying malicious web resources may also be covered by the following description. The goal of this is also exclusively to assist in understanding, and not to define the scope or limitations to these techniques. These modifications do not constitute an exhaustive list, and persons skilled in the art will recognize that other modifications are possible. Furthermore, it should not be interpreted in such a way that no modifications are possible where no examples of modifications have been stated, and/or that something described is the only option for implementing a certain element of these techniques. As will be clear to a person skilled in the art, this is most likely not the case.

Furthermore, it should be borne in mind that, for several specific manifestations, the method and system for identifying malicious web resources consist of simple different embodiments of these techniques, which in such cases are presented here in order to facilitate understanding. As a person skilled in the art will understand, many different embodiments of these techniques will possess much greater complexity.

Fig. 1 is a simplified schematic representation of one of the non-limiting embodiments of system 100 to implement these techniques.

In some non-limiting embodiments, system 100, which implements the method, comprises a data transfer network 102, at least one computer device 104 to identify malicious web resources (hereinafter computer device 104), processor 106 included in computer device 104, and database 108. System 100 may also include at least one search engine 110, which has access to web resource database 112.

In another non-limiting embodiment, system 100 may include additional active network equipment (not shown) and a client device (not shown).

Data transmission network 102 may be, for example, the Internet or any other computer network.

Computer device 104 may be an ordinary computer server. In the example embodiment, computer device 104 may be a Dell™ PowerEdge™ server, which uses the Ubuntu Server 18.04 operating system. Computer device 104 may be any other suitable hardware and/or application software and/or system software, or a combination thereof. In this non-limiting embodiment, computer device 104 is a single server. In other non-limiting embodiments, the functionality of the computer device may be divided among several computer devices, and, for example, may be provided by several servers.

Embodiments of computer device 104 are well known. Thus, it is sufficient to note that computer device 104 includes, among other things, a data transmission network interface (for example, a modem, network map, etc.) for two-way communication over a data transmission network (not shown); and processor 106, connected to the data transmission network interface, where processor 106 is configured to perform various procedures, including those described below. With this goal, processor 106 can have access to machine-readable instructions stored on machine-readable medium (not shown), the execution of which initiates processor 106's implementation of various procedures described here.

In some embodiments, computer device 104 includes one or several databases 108, which are configured to store at least some of the content element pointers, content elements, groups of content elements, and pointers to web resources associated with the relevant content element or group of content elements. In the context of this description, associated means linked or contained. For example, a web resource that has all the content elements from a group of content elements is associated with that group of content elements.

Although database 108 is represented as separate from computer device 104, to which it is connected using the corresponding line (not numbered), database 108 may be implemented as part of computer device 104 or will be available to computer device 104 by other means.

In another possible embodiment of system 100 for which computer device 104 includes machine-readable medium (not shown), on which machine-readable instructions and/or database 108 may be stored in advance. Alternatively or additionally, computer device 104 may have access to machine-readable instructions and/or database 108 located on other hardware. Thus, as implemented, processor 106 of computer device 104's access to machine-readable instructions and database 108 is not a limiting condition of these techniques.

Processor 106 is configured to receive pointers for a plurality of web resources. As a non-limiting example, receiving pointers for a plurality of web resources may entail receiving a list of uniform resource locators (URLs). The source or sources of pointers for a plurality of web resources may be various.

In another possible embodiment, in which processor 106 is configured to obtain pointers for the plurality of web resources at least partially from an external source via data network 102. For example, the external source could be search server 110, which has access to web resource database 112 and/or a server that collects data on malicious web resources (not shown), etc. Moreover, computer device 104 may have access to web resource database 112 via data transmission network 102. Alternatively or additionally, an external source may be a web resource, including pointers for a plurality of web resources.

Thus, exactly how and by what principle pointers for web resources are grouped on an external source, in particular, on web resource database 112, is not a limiting condition.

In the example embodiment, search engine 110 may be a Dell™ PowerEdge™ server, which uses the Ubuntu Server operating system. Search engine 110 also may be any other

suitable hardware and/or application software and/or system software, or a combination thereof. In this non-limiting embodiment, search engine 110 is a single server. In other non-limiting embodiments, the functionality of search engine 110 may be divided, and may be provided by several servers.

Embodiments of search engine 110 are well known. Thus, it is sufficient to note that search engine 110 includes, among other things, a data transmission network interface (for example, a modem, network card, etc.) for two-way communication over data transmission network 102; and a processor (not shown), connected to the data transmission network interface, the processor is configured to search web resources on data transmission network 102 and store them in database 112. With this goal, the search engine processor can have access to relevant machine-readable instructions stored on a machine-readable medium (not shown).

The process of data updating and saving in web resource database 112 is generally known as "data crawling". The search engine 110 is configured to "visit" various web resources, websites, and web pages that are accessible via data transmission network 102, and save them in web resource database 112 based on one or more predefined parameters. As a non-limiting example, server 110 can scan data transmission network 102 and save all new and updated web resources, thereby assembling a full web resource database for data transmission network 102, and/or collect web resources, for example, those including certain keywords in one or more languages. Since malicious web resources (in particular phishing websites) typically attempt to imitate bank websites, payment systems, and email services, some example keywords may be, without imposing limitations: "bank", "loan", "credit card", "mail", "account", "user name", "login", etc. The search can also be performed by domain names, and/or other parameters. Found and selected web resource pointers are saved in web resource database 112, after which they can be obtained by the processor of computer device 104 via data transmission network 102 from search engine 110.

In another possible embodiment, in which processor 106 of computer device 104 is configured to obtain pointers for the plurality of web resources directly from a previously generated web resource database 112. In this embodiment, the previously generated web resource database 112 may be implemented as part of computer device 104 or by accessible to computer device 104 by other means.

In another possible embodiment, in which processor 106 is configured to scan network 102 in order to obtain pointers for the plurality of web resources and in this way computer device

104 can perform the functionality of search engine 110 described above. For example, the scan of the network may be performed by web resource IP addresses, domain names, keywords, and other parameters.

Processor 106 of computer device 104 is configured to extract at least part of the content elements of each web resource from the plurality of web resources. Web resource content elements may be files of content element and/or hash sums of files of content elements (calculated, for example, with sha256 algorithm). The format of content element files is not specifically limited and depending on the specific type of content element. For example, in the case when the content element is an image, the file may be in one of the following formats: "jpg", "jpeg", "png", "bmp", "gif", etc. In the case when the content element is text, the file may be in one of the following formats: "txt", "doc", "html", etc. In the case when the content element is a font, the file may be in one of the following formats: "woff", "ttf", "eot", "svg", etc. In the case when the content element is a script, the file may be in one of the following formats: "asp", "aspx", "php", "jsp", "cgi", etc. In the case when the content element is animation or video, the file may be in one of the following formats: "flv", "swf", "avi", "mp4", "mov", etc. In the case when the content element is a cascading style sheet, the file may be in the format: "css". The specified list of possible content elements and format in which they may appear is not a limiting condition on these techniques.

The extraction of web resource content elements is performed by the processor 106. Extraction methods are well known, for example, a virtual browser such as Selenium may be used. The "cache all" option is set in the virtual browser, the cash is cleared, there is a transition to the web resource, after which the web resource's content elements are automatically saved in the browser's cache. Thus, content elements may be extracted from the plurality of web resources.

In another possible embodiment, when the pointers for the plurality of web resources are obtained, processor 106 is configured to obtain at least part of previously extracted pointers for content elements of the plurality of web resources.

In another possible embodiment, processor 106 is configured to perform filtration by deleting predefined standard content elements. During further processing, these filtered standard content elements are not taken into consideration. For example, cookies, common scripts, fonts, images, etc. may be predefined as standard content elements. The filtration criteria may be predefined by an expert or automatically, based on statistical data on the use of content elements

in a plurality of web resources. For example, when the predefined threshold for the quantity of web resources associated with a given content element is exceeded, it can be considered a standard content item. The threshold can be set manually by the operator, or using various automated algorithms, including algorithms based on machine learning.

Processor 106 is configured to combine at least two content elements into at least one first subgroup in response to the quantity of web resources that include said content elements exceeds a predefined minimum threshold of web resources for that subgroup.

Processor 106 is configured to combine at least two content elements into at least one second subgroup in response to the quantity of web resources that include said content elements exceeds a predefined minimum threshold of web resources for that subgroup.

Processor 106 is configured to combine the first and second subgroups into a group of content elements in response to the quantity of web resources that include the content elements of the first and second subgroups exceeds a predefined threshold of web resources for that subgroup. The process of combining content elements into subgroups and groups will be described in more detail below as part of the description of the method.

Processor 106 is also configured to save each group of content elements and pointers for web resources associated with the corresponding group of content elements in database 108.

Fig. 2 is an illustrative example of database 108, containing data on groups of content elements 202 and web resources 204 associated with the relevant group of content elements. In Fig. 2's illustrative example of database 108, only two groups of content elements are shown. One includes three content elements 410, 416, 418 associated with four web resources 400, 500, 600, and 700, and the second includes four content elements 406, 408, 418, 420 associated with three web resources 400, 500, and 700. Web resource examples 400, 500, 600, and 700 are shown in fig. 4, fig. 5, fig. 6, and fig. 7 respectively. It is worth noting that the quantity and quality of populating database 108 is not specifically limited, the purpose of these examples of groups of content elements 202 and web resources 204 is only to facilitate understanding these techniques, and they are not limiting. It is obvious that populating database 108 may entail much greater complexity. The quantity of groups of content elements 202 may be more than two, each group may contain at least three content elements, and different quantities of web resources 204 may be associated with each group of content elements 202. In this regard, the same web resource may be simultaneously associated with one or several groups of content elements 202.

In another possible embodiment, processor 106 is configured to identify malicious web resources associated with the group of content elements in database 108. For example, the identification of malicious web resources may be carried out by a machine-learning algorithm that has previously been trained in selecting malicious web resources. In another possible embodiment, the identification of malicious web resources is performed at least partially by a human expert. The human expert may, via a separate client device (not shown), have access to database 108 and verify web resources 204 associated with the groups of content elements 202.

Examples of client devices (among others) may be personal computers (desktops, laptops, netbooks, etc.), smartphones, tablets, as well as network equipment such as routers, switches, and gateways. It should be borne in mind that a device that behaves like a client device in this context may behave like a server in relation to other client devices. Use of the expression "client device" does not exclude the possibility of using a plurality of client devices to receive/send, execute, or initiate the execution of any task or request, nor the consequences of any task or request, nor the steps of any of the above-listed methods. A client device's communication with a user (human expert) does not imply any particular mode of operation, nor the need to login into a system, register, or anything similar. Accessing and exchanging data with database 108 and/or computer device 104 may be done by a client device (not shown), for example, via data transmission network 102 or another network (not shown).

Next, referring to Fig. 3, the described method 300 for identifying malicious web resources will be discussed in more detail; this method is performed by non-limiting embodiments of these techniques. Method 300 may be performed on computer device 104 and, specifically, by processor 106 of computer device 104 as in the non-limiting embodiment of system 100 in fig. 1.

Referring to Fig. 4, Fig. 5, Fig. 6, and Fig. 7, an illustrative example of implementing the method will be described below.

Fig. 4 shows an example of a first web resource 400, which includes content elements, in particular 402, 404, 406, 408, 410, 412, 414, 418, and 420. Furthermore, content elements 402, 410, 412, 414, and 418 are images, content element 404 is a banner with text, 406 is a user authorization form that includes several fillable fields, text, and a button, 408 is a menu element that includes an icon and text, 416 is text, and 420 is a web page menu script.

Fig. 5 shows an example of a second web resource 500, which includes content elements, in particular, 502, 504, 406, 408, 410, 412, 414, 416, 418, 420, and 512. Content elements 406,

408, 410, 412, 414, 416, 418, and 420 are also in the first web resource 400 and are described above. Content elements 502, 504 are images, content element 512 is a banner with text.

Fig. 6 shows an example of a third web resource 600, which includes content elements, in particular, 402, 404, 410, 416, 418, 420, 602, 606, 608, 610, and 612. Content elements 402, 404, 410, 416, 418, and 420 are also in the first web resource 400 and are described above. Content element 606 is a user authorization form that includes several fillable fields, text, and a button, content elements 608, 610, and 612 are images.

Fig. 7 shows an example of a fourth web resource 700, which includes content elements, in particular, 702, 704, 406, 408, 410, 412, 414, 416, 418, 420, and 504. Content elements 406, 408, 410, 412, 414, 416, 418, and 420 are also in the first web resource 400 and are described above. Content element 504 is also in the second web resource 500 and is described above. Content element 702 is an image, content element 704 is a banner with text.

In the context of this description, the quantity and types of different content elements are not limiting conditions. The web resources in Fig. 4, Fig. 5, Fig. 6, and Fig. 7 are given only as illustrative examples; these techniques may also be applied to any other web resources that possess a higher or lower quantity of content elements, where the types of content elements and even the content itself may be repeated or not repeated within the same web resource. Some of the content elements are not numbered in the illustrative web resource examples given.

The predefined minimum threshold for the quantity of web resources associated with a subgroup of content element is the minimum threshold of web resources for that subgroup. For this illustrative example, the minimum threshold of web resources for that subgroup is equal to 2. Furthermore, the predefined minimum threshold for the quantity of web resources associated with a group of content element is the minimum threshold of web resources for that group. For this illustrative example, let us assume that this minimum threshold of web resources for that group is also equal to 2. In another possible embodiment, the minimum threshold of web resources for that group is less than or equal to the minimum threshold of web resources for that subgroup. For example, the minimum threshold of web resources for that subgroup is equal to 150, but the minimum threshold of web resources for that group is equal to 10.

In another possible embodiment of method 300, there is a predefined maximum threshold for the quantity of web resources associated with a content element. For example, the threshold may be equal to 450, or to any other natural number.

In another possible embodiment of method 300, there are predefined threshold group weights, the group weights are determined by the ratio of the quantity of content elements in the group and the quantity of web resources associated with a given group of content elements. For example they may be defined by a function $\text{Score}(L, q)$, where L is the quantity of content elements in the group, and q is the quantity of web resources associated with the group. The function defining the weight of a group may be plurality by an expert (operator) and/or set by a machine-learning algorithm.

Here is an example of the function $\text{Score}(L, q)$ to define the group weights of content elements:

```
def score(L, q):
    if L <= 1:
        return 0.0
    if L >= 10:
        koef=2.0
    else:
        koef = {
            2: 0.5,
            3: 0.8,
            4: 1.0,
            5: 1.15,
            6: 1.3,
            7: 1.5,
            8: 1.7,
            9: 1.9,
        }[L]
    return q*koef
```

Stage 302 - receiving pointers for a plurality of web resources

Method 300 begins at step 302, where computer device 104 receives pointers for a plurality of web resources. Pointers for a plurality of web resources may be web resources themselves or links to web resources, URLs that are accessible via data transmission network 102 or another network (not shown) to computer device 104. In this regard, the plurality of web resources contains at least two web resources.

With regard to the non-limiting illustrative example, during step 302 computer device 104 receives pointers for the plurality of four web resources 400, 500, 600, and 700 from an external source - search engine 110 from web resource database 112. Web resources 400, 500, 600, and 700 were previously found by search engine 110 and saved in web resource database 112. Pointers for the plurality of web resources may be received at least partially from a previously composed web resource database 112. Also in another possible embodiment of method 300,

computer device 104 receives pointers for the plurality of web resource directly from web resource database 112.

In another possible embodiment of the method, computer device 104 performs a scan of data transmission network 102 in order to obtain the pointers for the plurality of web resources. The scan of data transmission network 102 may be performed by computer device 104 in the same way as for search engine 110 as described above, or in any other way.

In another possible embodiment of the method, during step 302, a plurality of web resources is received from at least two different sources. For example, web resources 400 and 600 are obtained via search engine 110 from web resource database 112, but web resources 500 and 700 are obtained from another external source (not shown) via data transmission network 102.

Method 300 then transits to performing step 304.

Stage 304 - extracting at least part of the content elements of each web resource in the plurality of web resources

During step 304 processor 106 of computer device 104 extracts at least part of the content elements from the plurality of web resources that were obtained during step 302. According to the non-limiting illustrative example, content elements 402, 404, 406, 408, 410, 412, 414, 418, and 420 are extracted from the first web resource 400, content elements 502, 504, 406, 408, 410, 412, 414, 416, 418, 420, and 512 are extracted from the second web resource 500, content elements 402, 404, 410, 416, 418, 420, 602, 606, 608, 610, and 612 are extracted from the third web resource 600, and content elements 702, 704, 406, 408, 410, 412, 414, 416, 418, 420, and 504 are extracted from the fourth web resource 700.

In another possible embodiment of the method, part of the content elements, or pointers for some of the content elements, are obtained during step 302 along with the plurality of web resources. For example, web resources and at least part of content elements extracted from the relevant web resources are stored in advance in web resource database 112.

In another possible embodiment of the method, files of content elements or hash sums of files of content elements are extracted as content elements.

Method 300 then transits to performing step 306.

Stage 306 - combining at least two content elements into at least one first subgroup in response to the quantity of web resources that include said content elements exceeds a predefined minimum threshold of web resources for that subgroup;

During step 306 processor 106 of computer device 104 creates all possible combinations of at least two different content elements. For each created combination, processor 106 checks for the presence of a predefined minimum quantity of web resources that include this combination of content elements. When the quantity of web resources that include said combination of at least two content elements exceeds the predefined minimum threshold, processor 106 combines at least two content elements into at least one first subgroup of content elements.

In this illustrative example, processor 106 creates the following combinations of content elements that have been extracted from web resources 400, 500, 600, and 700:

....

402-406

402-410

....

406-408

....

502-402

....

410-416

410-418

...

420-418

...

504-410

...

608-702-404

...

702-420

...

In this simplified illustrative embodiment, many different combinations of web resource content elements are possible, and the combinations given above are only a small portion of all possible combinations.

In this illustrative embodiment of method 300, the predefined minimum threshold of web resources for that subgroup is equal to 2. This criterion is met by, for example, the following pairs of elements: **406-408; 420-418.**

Processor 106 combines this pair of content elements into two first subgroups of content elements.

In another possible embodiment of the method, after extracting at least part of the content elements, a filtration is performed by deleting previously defined standard content elements, and then at least two elements are combined into at least one subgroup without the deleted content elements.

In another possible embodiment, there is a predefined maximum threshold for the quantity of web resources associated with a content element. For example, if this value is equal to 450 (or any other natural number n), then the content elements that belong to over 450 (over n) web resources are also excluded by processor 106, and not taken into consideration when creating the combinations of content elements extracted from web resources.

Stage 308 - combining at least two content elements into at least one second subgroup in response to the quantity of web resources that include said content elements exceeds a predefined minimum threshold;

During step 308, as in step 306 described above, processor 106 of computer device 104 combines at least two content elements into at least one second subgroup of content elements.

In the context of this description, the concepts of "first" and "second" subgroups of content elements are introduced solely to facilitate understanding, and to explain that these are not the same subgroup of content elements. It is worth noting that some content elements may coincide between the first and second subgroup of content elements.

The second subgroup of content elements is compiled by processor 106 in a similar way based on all possible combinations of at least two different content elements, as described above.

In this illustrative embodiment of method 300, the predefined minimum threshold of web resources is equal to 2. This criterion is also met by, for example, the following pairs of elements: **410-416; 410-418**, which are present in more than two web resources. Processor 106 combines this pair of content elements into a second subgroup of content elements.

Processor 106 can create a plurality of subgroups of content elements as was shown in the description of steps 306 and 308 of method 300. The shown embodiment, which includes two first and two second subgroups of content elements, is not limiting. The quantity of possible subgroups of content elements combined by processor 106 may be in the tens, hundreds, thousands, etc., depending on the quantity of received web resources and the quantity of extracted and coinciding content elements.

Method 300 then transits to performing step 310.

Stage 310 - combining the first and second subgroups into a group of content elements in response to the quantity of web resources that include the content elements of the first and second subgroups exceeds a predefined threshold of web resources for that subgroup;

During step 310 processor 106 of computer device 104 creates all possible combinations of at least two different subgroups of content elements obtained during steps 306 and 308. For each created combination, processor 106 checks for the presence of a predefined minimum quantity of web resources that include this combination of content elements. When the quantity of web resources that include the content elements of the first and second subgroups exceeds a predefined minimum threshold of web resources for that group, processor 106 combines at least one first subgroup of content elements and at least one second subgroup of content element into at least one group of content elements.

In this illustrative embodiment of method 300, there are only two first subgroups of content elements 406-408, 420-418 and two second subgroups of content elements 410-416; 410-418. The first subgroup of content elements 406-408 may be combined, for example, with the second subgroup of content elements 410-416 into a group of content elements, because all four content elements are in the first 400, second 500, and fourth 700 web resources, which exceeds the predefined threshold of web resources for that group of 2. Consequently, the first created group of content elements includes four content elements 406-408-410-416 associated with three web resources 400, 500, and 700.

Another first subgroup of content elements 420-418 may be combined, for example, with another second subgroup of content elements 410-418 into a group of content elements, because all these content elements are in the first 400, second 500, third 600, and fourth 700 web resources, which exceeds the predefined threshold of web resources for that group of 2. Consequently, the created second group of content elements includes three content elements 408-410-416 associated with four web resources 400, 500, 600, and 700.

The quantity of possible subgroups of content elements combined by processor 106 may be in the tens, hundreds, thousands, etc., depending on the quantity of received web resources and the quantity of extracted and coinciding content elements. The quantity of groups may be equal to one or more, despite the fact that two groups of content elements are shown in this illustrative example.

In another possible embodiment, prior to saving the group of content elements, the weight is determined for each group based on the ratio of the quantity of content elements in the group and the quantity of web resources associated with that group of content elements, the first and second subgroup of content elements are combined into a group in response the weight of the obtained group exceeds the predefined threshold.

In this illustrative example, the weight of the second group of content elements is equal to 3.2, and the weight of the first group of content elements is equal to 3. Taking into consideration the example of the Score (L,q) function given above, it exceeds the weight of the first group of content elements because it is smaller. In another possible embodiment of this method, the threshold weight of a group is equal to 3; in this case, during step 310, according to this illustrative example only the second group of content elements will be formed or created.

Method 300 then transits to performing step 312.

Stage 312 - saving the content elements group and the pointers for the web resources associated with the relevant content element group in the database.

During step 312, processor 106 of computer device 104 saves the group or groups of content elements created during step 310 in database 108. Processor 106 also saves pointers for web resources associated with the relevant group or groups of content elements. In this case, web resources that include all content elements of the group of content elements are considered to be associated with the group of content elements.

In this illustrative embodiment of method 300, the processor 106 saves in database 108 the first group of content elements that includes content elements 406-408-410-416, with which three web resources 400, 500, and 700 are associated, and the second group of content elements that includes three content elements 410-418-420, with which four web resources 400, 500, 600, and 700 are associated.

Then, the execution of method 300 may be terminated.

In another possible embodiment, after saving each group of content elements and pointers for web resources associated with said group of content elements in the database, pointers are obtained for at least one additional web resource; at least part of the content elements of each web resource is extracted; in response to an identification in the database of at least one group of content elements with which said at least one specified additional web resource is associated, the pointers for the additional web resource are saved in the database.

In another possible embodiment of the method, after saving the group of content elements and pointers for web resources associated with said group of content elements, malicious web resources associated with the group of content elements are also identified. In another possible embodiment of the method in which the identification of malicious web resources is performed at least partially by a human expert.

The claimed method for identifying malicious web resources ensures the automatic clustering of pluralities of web resources into groups, which allows an expert to perform a check in real time on each of the saved groups, and to identify groups of phishing sites, for example, those created by phishing kits, and also to identify mirrors and copies of blocked and previously identified malicious web resources. In this way, the expert's workload is substantially reduced during the process of identifying malicious web resources.

Modifications and improvements to the above-described embodiments will be clear to persons skilled in the art. The preceding description was given only as an example and does not impose any limitations. Thus, the scope of these techniques is limited only by the scope of the appended claims.

CLAIMS

1. A method for identifying malicious web resources, executed on a computer device, the method comprising:

receiving pointers for a plurality of web resources;

extracting at least some of the content elements of each web resource in the plurality of web resources;

combining at least two content elements into at least one first subgroup in response to the quantity of web resources including said content elements exceeds a predefined minimum threshold of web resources for that subgroup;

combining at least two content elements into at least one second subgroup in response to quantity of web resources including said content elements exceeds a predefined minimum threshold of web resources for that subgroup;

combining the first and second subgroups into a group of content elements in response to the quantity of web resources including the content elements of the first and second subgroups exceeds a predefined threshold of web resources for that subgroup;

saving the content element group and the pointers for the web resources associated with the relevant content element group in the database.

2. The method of claim 1, wherein the plurality of web resources comprises at least two web resources.

3. The method of claim 1, wherein a preliminary network scan is performed to obtain pointers for a plurality of web resources.

4. The method of claim 1, wherein the pointers for the plurality of web resources are obtained at least partially from an external source via a data network.

5. The method of claim 1, wherein the pointers for the plurality of web resources are obtained from a previously composed database of web resources.

6. The method of claim 1, wherein at the step of receiving the pointers for the plurality of web resources at least some pointers are obtained from previously extracted pointers for content elements of the plurality of web resources.

7. The method of claim 1, wherein the content elements are files of content elements or hash sums of files of content elements.
8. The method of claim 1, wherein after extracting at least some of the content elements, a filtration is performed by deleting of previously defined standard content elements, and at least two elements are combined into at least one subgroup without the deleted content elements.
9. The method of claim 1, wherein prior to saving the group of content elements, the weight is determined for a group based on the ratio of the quantity of content elements in the group and the quantity of web resources associated with that group of content elements, the first and second subgroup of content elements are combined into a group in response to the weight of the obtained group exceeds the predefined threshold.
10. The method of claim 1, wherein after saving the group of content elements and pointers for web resources associated with said group of content elements, malicious web resources associated with the group of content elements are also identified.
11. The method of claim 1, wherein the identification of malicious web resources is performed at least partially by a human expert.
12. A computer device that identifies malicious web resources, connected to a data network and at least one database, including a computer processor, the computer processor is configured to:

receive pointers for a plurality of web resources

extract at least some of the content elements of each web resource in the plurality of web resources

combine at least two content elements into at least one first subgroup in response to the quantity of web resources including said content elements exceeds a predefined minimum threshold of web resources for that subgroup

combine at least two content elements into at least one second subgroup in response to the quantity of web resources including said content elements exceeds a predefined minimum threshold of web resources for that subgroup

combine the first and second subgroups into a group of content elements in response to the quantity of web resources including the content elements of the first and second subgroups exceeds a predefined threshold of web resources for that subgroup

save the content element group and the pointers for the web resources associated with the relevant content element group in the database.

13. The device of claim 12, wherein the processor is configured to perform a network scan to obtain pointers for a plurality of web resources.
14. The device of claim 12, wherein the processor is configured to obtain pointers for the plurality of web resources at least partially from an external source via a data network.
15. The device of claim 12, wherein the processor is configured to obtain pointers for the plurality of web resources from a previously composed database of web resources.
16. The device of claim 12, wherein at the step of receiving the pointers for the plurality of web resources, the processor is configured to obtain at least some from previously extracted pointers for content elements of the plurality of web resources.
17. The device of claim 12, wherein the processor is configured to extract at least some of the content elements of the web resource when the content elements are files of content elements or hash sums of files of content elements.
18. The device of claim 12, wherein, after extracting at least some of the content elements, the processor is configured to perform a filtration by deleting previously defined standard content elements, and to combine at least two elements into a group without the deleted content elements.