

METHOD AND APPARATUS FOR AUTHENTICATING A WEBSITE

FIELD OF THE INVENTION

The present invention relates to communications techniques, and more particularly, to a method and an apparatus for recognizing a website.

BACKGROUND OF THE INVENTION

With the booming of the Internet, various kinds of websites make people's daily life rich and convenient. However, there also come along some insecure websites, e.g., phishing websites. They are fraudulent websites attempting to obtain users' sensitive information, such as user name, password, account and credit card details to defraud the users of their possessions. There are also some virus websites which broadcast virus when users visit them. Therefore, there is a need to determine whether a website is secure when people visit the website.

In the prior art, one existing method recognizes a phishing website through finding suspicious feature of a Uniform Resource Locator (URL) of the website. For example, suppose the true URL of a bank is <http://www.wmd.com>. A counterfeit URL of the website may be <http://www.wmd.com.mn> or <http://www.wmd.com%2e%61%62%63%2e%63%6f%6d> which is obtained by coding the true URL of the bank. Although what is seen by the user is wmd.com, the browser will decode contents containing “%” according to URL coding rules to obtain a real destination address <http://www.wmd.com.abc.com>. The true destination domain name is abc.com but not wmd.com. Through comparing the destination domain names, it is possible to recognize the phishing website. Another method recognizes the phishing website through maintaining a blacklist, wherein URLs in the blacklist are reported by users, or the URLs are URLs broadcasted for a large number of times by a suspicious source and are listed into the blacklist by finding the suspicious source.

The inventor of the present invention finds that the existing methods have at least the following defects.

The method which recognizes the phishing website through finding suspicious

feature of the URL requires that the URL contains the suspicious feature, e.g. coding symbol “%”, or requires that the URL partially overlaps with a known URL. Thus, if a URL does not contain suspicious code or does not overlap with known URLs, this method cannot recognize the website effectively. Therefore, there are limitations in this method.

As to the method maintaining the blacklist, the suspicious URL needs to be recognized and reported by users or needs to be found by searching for suspicious sources of URLs broadcasted for a large number of times before being listed in the blacklist. This method cannot recognize phishing websites timely and cannot recognize some phishing websites which are not broadcasted for many times but are seriously risky.

SUMMARY OF THE INVENTION

Embodiments of the present invention provide a method and an apparatus for recognizing a website, so as to recognize the website timely and effectively.

According to one aspect of the present invention, a method for recognizing a website is provided. The method includes:

- parsing a Document Object Model (DOM) tree corresponding to page code of a website to be recognized;

- comparing the DOM tree obtained by parsing with pre-stored DOM trees;

- recognizing the website to be recognized according to a compared result.

According to another aspect of the present invention, an apparatus for recognizing a website is provided. The apparatus includes:

- a parsing module, adapted to parse a Document Object Model (DOM) tree corresponding to page code of a website to be recognized;

- a comparing module, adapted to compare the DOM tree obtained by parsing with pre-stored DOM trees; and

- a recognizing module, adapted to recognize the website to be recognized according to compared result.

Advantages of the technical solution provided by the embodiments of the present invention are as follows.

Through parsing the DOM tree corresponding to the page code of the website to be recognized and comparing the DOM tree of the website to be recognized with the pre-stored DOM trees, it is possible to recognize the website to be recognized timely and effectively according to the compared result.

BRIEF DESCRIPTION OF THE DRAWINGS

Drawings used in the description of the embodiments will be introduced briefly hereinafter to clearly describe the technical solution provided by the embodiments of the present invention. Obviously, the following drawings are merely some examples. And other drawings can be obtained by those with ordinary skill in the art based on these drawings without inventive work.

FIG.1 is a flowchart illustrating a method for recognizing a website according to a first embodiment of the present invention.

FIG.2 is a flowchart illustrating a method for recognizing a website according to a second embodiment of the present invention.

FIG.3 is a schematic diagram illustrating HTML code and a DOM tree provided by the second embodiment of the present invention.

FIG.4 is a schematic diagram illustrating a first apparatus for recognizing a website according to a third embodiment of the present invention.

FIG.5 is a schematic diagram illustrating a second apparatus for recognizing a website according to the third embodiment of the present invention.

FIG.6 is a schematic diagram illustrating a third apparatus for recognizing a website according to the third embodiment of the present invention.

FIG.7 is a schematic diagram illustrating a fourth apparatus for recognizing a website according to the third embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Hereinafter, the present invention will be described in further detail with reference to accompanying drawings and embodiments to make the technical solution and merits therein clearer.

A first embodiment

The first embodiment provides a method for recognizing a website. As shown in **FIG.1**, the method includes the following steps.

Step **101**, parse a Document Object Model (DOM) tree corresponding to page code of a website to be recognized.

Step **102**, compare the DOM tree with pre-stored DOM trees.

Step **103**, recognize the website to be recognized according to the compared result.

Through parsing the DOM tree corresponding to the page code of the website to be recognized and comparing the DOM tree of the website to be recognized with the pre-stored DOM trees, it is possible to recognize the website to be recognized timely and effectively according to the compared result.

A second embodiment

The second embodiment provides a method for recognizing a website. Within certain time period, contents displayed by a website will change. But the layout of the website remains unchanged or only has little changes. For example, as to a website, the navigation bar is located in the same place today and yesterday. The position used for display weather forecast yesterday is still used for displaying weather forecast today. The positions of channel columns are also unchanged. What are changed are merely the number of links and titles in the channels columns. Therefore, most phishing websites visually resemble the corresponding true websites. The visual and layout information of a website is saved in the page code of the website. And the page code of the website can be parsed into a Document Object Model (DOM) tree. Take the parsing of Hyper Text Mark-up Language (HTML) code into a DOM tree as an example. A website browser

obtains the DOM tree through parsing the HTML code, displays page information to terminal users and performs page operation interactions with the terminal users based on nodes of the DOM tree and their values. Similarly, through parsing Extensible Markup Language (XML) code and Extensible Hyper Text Markup Language (XHTML) code into the DOM tree, it is also possible to recognize the website. The method provided by the embodiments of the present invention determines which secure website that the structure or layout of the website to be recognized resembles timely according to visual similarity of the websites, i.e. the similarity of the two DOM trees, so as to determine whether the website to be recognized is secure.

For facilitating the description, in this embodiment, suppose the website to be recognized (the website that the user wants to open) is website A, and website A is a phishing website of a known website B. The method for recognizing the website provided by the embodiments of the present invention will be described hereinafter taking the HTML language as an exemplary page code. As shown in **FIG.2**, the method includes the following steps.

Step **201**, before website A is open, a website recognizing apparatus retrieves HTML code corresponding to website A from a corresponding server according to an address of website A.

The address is an Internet Protocol (IP) address or a domain name address. Before the page code corresponding to the website to be recognized is retrieved from the corresponding server according to the domain name address, the domain name address needs to be translated into an IP address by a Domain Name Server (DNS). Then, the server can be found through the IP address. When the page code corresponding to the website to be recognized is retrieved from the server according to IP address, the process of translating the domain name by the DNS can be omitted. Therefore, in practical applications, it is more convenient to use IP address in this step. Whatsoever, this embodiment does not restrict the type of the address.

Step **202**, parse a DOM tree corresponding to the HTML code retrieved.

Specifically, **FIG.3** provides an instance for parsing HTML code into a DOM tree. As shown in **FIG.3**, the left is the HTML code of the website and the right is the DOM tree. According to definitions of the HTML, contents in parentheses in the left are labels.

A label usually starts with <label name> and ends with </label name>. A plurality of labels may be nested in one label structure. The type of the label may determine the layout of the website. For example, the label TABLE (table) in **FIG.3** is presented as a table in the webpage. In practical applications, there are many types of labels. For example, an INPUT label not mentioned in **FIG.3** will be presented as an input box for inputting user name or password.

When the HTML code is converted into the DOM tree, the structure and the order of nodes denote a visual layout of the website and values of the nodes denote contents seen by the users. As shown by the right part of **FIG.3**, TABLE is a root node. A child node of TABLE is TBODY (table body). TBODY has two child nodes. They are both TRs (table rows) and respectively denote one row in the table. TR has two child nodes and they are both TDs (table trades) and each of them represents one column of the table. The values of the TD nodes are respectively texts of temperature, time, city and region. These texts are contents seen by the users in the table of the webpage.

Step **203**, compare the DOM tree with pre-stored DOM trees.

In this embodiment, DOM trees corresponding to page codes of secure websites are taken as exemplary pre-stored DOM trees. With respect to insecure phishing websites, i.e. risky websites, the secure websites include but are not limited to websites requiring inputting of username, password or other account information for logon and having the risk of being counterfeited under profit-driven, e.g., bank websites, etc. In the website recognizing apparatus provided by the embodiments of the present invention, not only HTML code and a DOM tree corresponding to the HTML code of the true website corresponding to the phishing website are stored, the number of nodes in the DOM trees corresponding to the true website is also stored. In order to ensure the complete and accuracy of the DOM trees of the secure websites pre-stored, the DOM trees of the websites will be refreshed every certain period (e.g., every 5 seconds).

Specifically, the DOM tree has a depth direction (longitudinal direction) and a span direction (latitudinal direction). When comparing the DOM tree obtained by parsing with the pre-stored DOM trees, the embodiment of the present invention provides two comparing methods which are respectively corresponding to situations that the nodes of the DOM tree are weighted or not. On the one hand, the situation that

the nodes of the DOM tree are not weighted is firstly considered.

Firstly, among the pre-stored DOM trees, determine whether there is a DOM tree having the same nodes with the DOM tree obtained by parsing. If there is such a DOM tree, traverse the pre-stored DOM tree and the DOM tree obtained by parsing in the depth direction. Determine how many same nodes that the two DOM trees have and obtain the number of same nodes. Suppose the DOM tree of website A to be recognized is a, and the website recognizing apparatus stores DOM trees of a plurality of websites including the DOM tree b of the known website B. Then, the website to be recognized is very possible to be a counterfeit of a non-phishing website (i.e. a phishing website). If a ratio of the number of the same nodes m to the total number of nodes n of the DOM tree b corresponding to website B reaches a threshold, determine that the DOM tree obtained by parsing matches the pre-stored DOM tree. The threshold is able to reflect the similarity degree of the two websites. If they are not the same website, the higher the threshold, the higher the possibility that the website to be recognized being a counterfeit website, i.e. the higher the possibility that the website to be recognized being a phishing website. This embodiment does not restrict the detailed value of the threshold. For example, it is possible to set when $m/n \geq 60\%$, determine that the website to be recognized visually resembles the pre-stored non-phishing website.

On the other hand, in the page code, the type of the labels can determine the layout of the webpage. After the page code is parsed into the DOM tree, the nodes and their types and sequence in the DOM tree may reflect the visual layout of the website. Therefore, both the type and the sequence of the nodes can affect the visual layout of the website. Different types of nodes have different affections to the visual layout of the website. For example, a phishing website has nodes including account input box, password input box and tables which reflect main features of the website. Considering the affects of the types of the nodes to the visual layout, this embodiment weights the nodes of the DOM tree.

When the nodes of the DOM tree are weighted, the website recognizing apparatus sets different weights for different nodes of the DOM tree according to different types of the nodes. The higher the weight, the more important the node and the larger the affect the node has to the layout, i.e. the more relative the node to the feature of obtaining user account and password. For example, weights α of different nodes are

listed as follows from high to low: the weight of an input node in the html is 2.0, i.e. [input: 2.0], the weight of a root node in the html is 1.8, i.e. [table: 1.8], the weight of a child node of TABLE in the html is 1.8, i.e. [tbody: 1.8], the weight of a first label body in the html is 1.8, i.e. [body: 1.8], the weight of a image label node img in the html is 1.5, i.e. [img: 1.5], the weight of an object node in the html is 1.5, i.e. [object: 1.5], the weight of a tr node in the html is 1.5, i.e. [tr: 1.5], the weight of a td node in the html is 1.4, i.e. [td: 1.4], the weight of a block element node div in the html is 1.0, i.e. [div: 1.0], the weight of a paragraph node p in the html is 0.8, i.e. [p: 0.8], a br node in the html is 0.7, i.e. [br: 0.7], the weight of an inline element node span in the html code is 0.6, i.e. [span: 0.6]. Still suppose that there is a DOM tree A among the stored DOM trees which has the same nodes with the DOM tree obtained by parsing. Traverse in the depth direction the stored DOM tree A and the DOM tree obtained by parsing, determine how many same nodes that the two DOM trees have to obtain weight x of all the same nodes. Traverse the DOM tree A in the depth direction to obtain weight of all the same nodes. If the ratio of weight x of the same nodes to the total number of nodes n of the DOM tree A reaches a threshold, determine that the two DOM trees match. For example, suppose that when $x/n \geq 60\%$, it is determined that the website to be recognized visually resembles the stored non-phishing website, i.e. the two DOM trees match. This embodiment does not restrict the weights set to different nodes.

Step **204**, after finding the DOM tree matches the DOM tree obtained by parsing, compare addresses corresponding to the two DOM trees, if they are the same, proceed to step **205**, otherwise, proceed to step **206**.

Step **205**, determine that the website to be recognized is not a phishing website, and the procedure ends.

Step **206**, determine that the website to be recognized is a phishing website, and prevent users from browsing the phishing website.

This embodiment only takes the situation that the pre-stored DOM trees are those corresponding to secure websites as an example. Similarly, the pre-stored DOM trees may also be DOM trees of risky websites.

With respect to this situation, after comparing the DOM tree of the website to be recognized with the pre-stored DOM trees and finding a matching DOM tree, if the

addresses corresponding to the two DOM trees are the same, it is determined that the website to be recognized is a risky website. If the two DOM trees are different, it is determined that the website to be recognized is a secure website. In practical applications, even if the pre-stored DOM trees are updated every certain time period, it is hard to cover the DOM trees of all risky websites. Therefore, even if the address of the website to be recognized is not the same with those of the stored risky websites, the website to be recognized may not be a secure website. Therefore, this method has a low reliability and is not very secure.

As to the situation that the pre-stored DOM trees are the DOM trees of the secure websites, although there may be error in the recognized result, this manner will not recognize a risky website as a secure website and thus has a relatively high reliability and high security.

In view of the above, the method provided by the embodiment of the present invention retrieves the HTML code of the website to be recognized, parses the DOM tree corresponding to the HTML code of the website to be recognized, compares the DOM tree of the website to be recognized with those corresponding to the HTML code of non-phishing websites pre-stored, determines whether the website to be recognized is a phishing website according to a compared result and the addresses of the website to be recognized and the non-phishing websites. The method can recognize the phishing website timely and effectively, and can prevent users from browsing the phishing website after recognizing the phishing website, which prevents the users from being cheated and safeguards the user's profit.

A third embodiment

This embodiment provides an apparatus for recognizing a website. As shown in **FIG.4**, the apparatus includes:

a parsing module **401**, adapted to parse a Document Object Model (DOM) tree corresponding to page code of a website to be recognized;

a comparing module **402**, adapted to compare the DOM tree with pre-stored DOM trees; and

a recognizing module **403**, adapted to recognize the website to be recognized

according to a compared result.

Specifically, the recognizing module **403** is adapted to determine the website to be recognized as a risky website if there is a matching DOM tree among the DOM trees matching the DOM tree obtained by parsing and the address of the website to be recognized is not the same as the address corresponding to the matching DOM tree.

As shown in **FIG.5**, as to the situation that the nodes in the DOM tree are not weighted, the comparing module **402** specifically includes:

a first determining unit **402a**, adapted to determine, among the stored DOM trees, whether there is a DOM tree having same nodes with the DOM tree obtained by parsing, and take the DOM tree having same nodes with the DOM tree obtained by parsing as the matching DOM tree;

a first traversing unit **402b**, adapted to traverse in the depth direction the matching DOM tree and the DOM tree obtained by parsing to obtain the number of same nodes; and

a first matching determining unit **402c**, adapted to determine whether a ratio of the number of same nodes to the total number of nodes of the matching DOM tree reaches a threshold, and determine that the matching DOM tree matches the DOM tree the if the ratio reaches the threshold.

As shown in **FIG.6**, as to the situation that the nodes in the DOM tree are weighted, the comparing module **402** specifically includes:

a weighting unit **402d**, adapted to set a weight for each node in the DOM trees pre-stored;

a second determining unit **402e**, adapted to determine, among the stored DOM trees, whether there is a DOM tree having same nodes with the DOM tree obtained by parsing, and take the DOM tree having same nodes with the DOM tree obtained by parsing as the matching DOM tree;

a second traversing unit **402f**, adapted to traverse in the depth direction the matching DOM tree and the DOM tree obtained by parsing to obtain a weight of all the

same nodes; and

a second matching determining unit **402g**, adapted to determine whether a ratio of the weight of all the same nodes to the total number of nodes of the matching DOM tree reaches a threshold, and determine that the matching DOM tree matches the DOM tree obtained by parsing if the ratio reaches the threshold.

Furthermore, as shown in **FIG.7**, the apparatus may further include:

a retrieving module **404**, adapted to retrieve page code of the website to be recognized according to an address of the website to be recognized.

In view of the above, the method provided by the embodiment of the present invention retrieves the HTML code of the website to be recognized, parses the DOM tree corresponding to the HTML code of the website to be recognized, compares the DOM tree of the website to be recognized with those corresponding to the HTML codes of non-phishing websites pre-stored, determines whether the website to be recognized is a phishing website according to a compared result and the addresses of the website to be recognized and the non-phishing websites. The method can recognize the phishing website timely and effectively, and can prevent the users from browsing the phishing website after recognizing the phishing website, which prevents the users from being cheated and safeguards the user's profit.

The above sequential numbers of the embodiments are merely used for facilitating the description but not used for denoting the preference of the embodiments.

Some steps in the embodiments of the present invention may be implemented by software stored in a computer-readable medium, e.g. CD or hard disk.

The foregoing descriptions are only preferred embodiments of this invention and are not for use in limiting the protection scope thereof. Any changes and modifications can be made by those skilled in the art without departing from the spirit of this invention and therefore should be covered within the protection scope as set by the appended claims.

CLAIMS

1. A method for recognizing a website, comprising:

parsing a Document Object Model (DOM) tree corresponding to page code of a website to be recognized;

comparing the DOM tree obtained by parsing with pre-stored DOM trees;

recognizing the website to be recognized according to a compared result.

2. The method of claim 1, wherein the recognizing the website to be recognized according to the compared result comprises:

if there is a matching DOM tree among the pre-stored DOM tree and an address of the website to be recognized is different from an address of the matching DOM tree, determining that the website to be recognized is a risky website.

3. The method of claim 1, wherein the comparing the DOM tree obtained by parsing with the pre-stored DOM trees comprises:

determining, among the pre-stored DOM trees, whether there is a DOM tree having same nodes with the DOM tree obtained by parsing, if there is such a DOM tree, taking the DOM tree having same nodes with the DOM tree obtained by parsing as a matching DOM tree, traversing in a depth direction the matching DOM tree and the DOM tree obtained by parsing to obtain a number of same nodes;

if a ratio of the number of same nodes to a total number of nodes of the matching DOM tree reaches a threshold, determining that the matching DOM tree matches the DOM tree obtained by parsing.

4. The method of claim 1, wherein the comparing the DOM tree with the pre-stored DOM trees comprises:

weighting each node of the pre-stored DOM trees, determining, among the pre-stored DOM trees, whether there is a DOM tree having same nodes with the DOM tree obtained by parsing, taking the DOM tree having same nodes with the DOM tree obtained by parsing as a matching DOM tree, traversing, in a depth direction, the DOM

tree obtained by parsing and the matching DOM tree to obtain a weight of the same nodes; and

if a ratio of the weight of the same nodes to a total number of nodes of the matching DOM tree reaches a threshold, determining that the matching DOM tree matches the DOM tree obtained by parsing.

5. The method of anyone of claims 1 to 4, wherein the pre-stored DOM trees are corresponding to page codes of secure websites pre-stored and are refreshed every pre-defined time period.

6. The method of claim 1, wherein the pre-stored DOM trees are corresponding to page codes of risky websites pre-stored.

7. An apparatus for recognizing a website, comprising:

a parsing module, adapted to parse a Document Object Model (DOM) tree corresponding to page code of a website to be recognized;

a comparing module, adapted to compare the DOM tree obtained by parsing with pre-stored DOM trees; and

a recognizing module, adapted to recognize the website to be recognized according to compared result.

8. The apparatus of claim 7, wherein the recognizing module is further adapted to determine that the website to be recognized is a risky website if the there is a matching DOM tree among the DOM trees pre-stored and an address of the website to be recognized is different from that of the matching DOM tree.

9. The apparatus of claim 7, wherein the comparing module further comprises:

a first determining unit, adapted to determine, among the pre-stored DOM trees, whether there is a DOM tree having same nodes with the DOM tree obtained by parsing, taking the DOM tree having same nodes with the DOM tree as a matching DOM tree if there is such a DOM tree;

a first traversing unit, adapted to traverse, in a depth direction, the matching DOM

tree and the DOM tree obtained by parsing to obtain a number of same nodes;

a first matching determining unit, adapted to determine whether a ratio of the number of same nodes to a total number of nodes of the matching DOM tree reaches a threshold, and determine that the matching DOM tree matches the DOM tree obtained by parsing if the ratio reaches the threshold.

10. The apparatus of claim 7, wherein the comparing module further comprises:

a weighting unit, adapted to set a weight for each node of the DOM trees pre-stored;

a second determining unit, adapted to determine, among the stored DOM trees, whether there is a DOM tree having same nodes with the DOM tree obtained by parsing, and take the DOM tree having same nodes with the DOM tree obtained by parsing as a matching DOM tree;

a second traversing unit, adapted to traverse, in a depth direction, the matching DOM tree and the DOM tree obtained by parsing to obtain a weight of all the same nodes; and

a second matching determining unit, adapted to determine whether a ratio of the weight of all the same nodes to a total number of nodes of the matching DOM tree reaches a threshold, and determine that the matching DOM tree matches the DOM tree obtained by parsing if the ratio reaches the threshold.

11. The apparatus of claim 7, further comprising:

a retrieving module, adapted to retrieve the page code of the website to be recognized according to an address of the website to be recognized.