

A METHOD AND SYSTEM FOR MONITORING WEBSITE DEFACEMENTS

FIELD OF THE INVENTION

[0001] The invention pertains to the field of monitoring website defacements.

BACKGROUND

[0002] An increasing number of website defacement incidents are being reported. These defacements are the work of hackers, who break into a web server and replace the website with one of their own. Typically, website defacement incidents are attributed to politically motivated protestors and people bearing grudges against an organization or administration. Most times, the defacement is harmless and is only done to show off a hacker's skills. However, it can sometimes be used as a distraction to cover up more sinister actions such as uploading malware (malicious software).

[0003] US patent publication 2013/0097702 describes a website defacement system that receives web page information and snapshot images corresponding to websites and performs comparisons against corresponding information and snapshot images of a reference website. Probability scores indicating the likelihood that a website has been defaced are calculated based on the comparisons. However, US 2013/0097702 is silent on partitioning the snapshot image into smaller regions, allowing users to select from these partitioned regions, and carrying out image comparisons on these user-selected partitioned regions. Further, US 2013/0097702 is silent on describing how the image comparison is actually carried out.

[0004] It is therefore an object of an invention to solve the above deficiencies and at least to provide a novel method and system for monitoring website defacements.

SUMMARY OF INVENTION

[0005] According to a first aspect of the invention, a method for monitoring a website for defacement is described, the method comprising the steps of obtaining a baseline image of the website; partitioning the baseline image into a plurality of baseline image regions according to a plurality of partitions in a partitioning algorithm and allowing the partitions to

be selected. The method further comprises the steps of obtaining the selected partitions; storing the baseline image regions which correspond to the selected partitions in a database; obtaining a image instance of the website at a polled interval and partitioning the image instance into a plurality of image instance regions according to the plurality of partitions in the partitioning algorithm. The method further comprises the steps of extracting the image instance regions which correspond to the selected partitions; performing image comparison on the stored baseline image regions and the extracted image instance regions and sending an alert that the website has been defaced when a result of the image comparison exceeds a first threshold.

[0006] Preferably each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a image intensity value, and preferably the step of performing image comparison on the stored baseline image regions and the extracted image instance regions comprises the steps of calculating a first image intensity value for each of the stored baseline image regions by totaling up the image intensity values of the pixels; calculating a second image intensity value for each of the extracted image instance regions by totaling up the image intensity values of the pixels, and wherein the result of the image comparison is dependent on the difference between the first image intensity value and the second image intensity value.

[0007] Preferably each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a red channel value, a green channel value and a blue channel value, and preferably the step of performing image comparison on the stored baseline image regions and the extracted image instance regions comprises the steps of calculating a first red channel value for each of the stored baseline image regions by totaling up the red channel values of each pixel and calculating a second red channel value for each of the extracted image instance regions by totaling up the red channel values of each pixel. The method further comprises the steps of calculating a first green channel value for each of the stored baseline image regions by totaling up the green channel values of each pixel; calculating a second green channel value for each of the extracted image instance regions by totaling up the green channel values of each pixel; calculating a first blue channel value for each of the stored baseline image regions by totaling up the blue channel values of each pixel; calculating a second blue channel value for each of the extracted image instance regions by totaling up the blue channel values of each pixel, and wherein the result of the image

comparison is dependent on the difference between the first red channel value and the second red channel value, and on the difference between the first green channel value and the second green channel value, and on the difference between the first blue channel value and the second blue channel value.

[0008] Preferably each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a image intensity value, a red channel value, a green channel value and a blue channel value, and preferably the step of performing image comparison on the stored baseline image regions and the extracted image instance regions comprises the steps of comparing the image intensity value of each pixel of the stored baseline image regions with the image intensity value of each pixel of the extracted image instance regions to determine a number of pixels whose image intensity value has changed and comparing the red channel value of each pixel of the stored baseline image regions with the red channel value of each pixel of the extracted image instance regions to determine a number of pixels whose red channel value has changed. The method further comprises comparing the green channel value of each pixel of the stored baseline image regions with the green channel value of each pixel of the extracted image instance regions to determine a number of pixels whose green channel value has changed; comparing the blue channel value of each pixel of the stored baseline image regions with the blue channel value of each pixel of the extracted image instance regions to determine a number of pixels whose blue channel value has changed; and wherein the result of the image comparison is dependent on the number of pixels whose image intensity value has changed, and on the number of pixels whose red channel value has changed, and on the number of pixels whose green channel value has changed and on the number of pixels whose blue channel value has changed.

[0009] Preferably, the method further comprises the steps of obtaining a baseline HTML content of the website; storing the baseline HTML content in the database; obtaining a HTML content instance of the website at the polled interval; performing content comparison on the stored baseline HTML content and the HTML content instance; and sending an alert that the website has been defaced when a result of the content comparison exceeds a second threshold.

[0010] Preferably, the step of performing content comparison on the stored baseline HTML content and the HTML content instance comprises the steps of counting a number of

links in the stored baseline HTML content and the HTML content instance; counting a number of scripts in the stored baseline HTML content and the HTML content instance; and counting a number of images in the stored baseline HTML content and the HTML content instance.

[0011] Preferably, the method further comprises the steps of performing a first integrity comparison on the baseline image and the image instance; performing a second integrity comparison on the baseline HTML content and the HTML content instance; and sending an alert that the website has been defaced when a result of the first integrity comparison and a result of the second integrity comparison exceeds a third threshold.

[0012] Preferably, the step of performing a first integrity comparison on the baseline image and the image instance comprises the steps of hashing a image in the baseline image to obtain a first hash value; hashing a image in the image instance to obtain the second hash value; and comparing the first hash value and the second hash value.

[0013] Preferably, the step of performing a second integrity comparison on the baseline HTML content and the HTML content instance comprises the steps of hashing a script in the baseline HTML content to obtain a first hash value; hashing a script in the HTML content instance to obtain the second hash value; and comparing the first hash value and the second hash value.

[0014] Preferably, the method further comprises the steps of checking the HTML content instance for malwares and sending an alert that the website has been defaced when at least one malware is detected.

[0015] Preferably, the method further comprises the steps of waiting for a predetermined period to lapse after obtaining the baseline HTML content of the website; obtaining another baseline HTML content of the website, and comparing the baseline HTML content with the another baseline HTML content and allowing the second threshold and third threshold to be adjusted based on this comparison.

[0016] Preferably, the partitioning algorithm is anyone of the following: a 4 by 4 grid; a 3 by 3 grid, a 5 by 5 grid and a 6 by 6 grid.

[0017] According to a second aspect of the invention, a system for monitoring a website for defacement is described, comprising a database and at least one processor programmed to obtain a baseline image of the website; partition the baseline image into a plurality of baseline image regions according to a plurality of partitions in a partitioning algorithm; allow the partitions to be selected and obtain the selected partitions. The processor is further programmed to store the baseline image regions which correspond to the selected partitions in the database; obtain a image instance of the website at a polled interval and partition the image instance into a plurality of image instance regions according to the partitions. The processor is further programmed to extract the image instance regions which correspond to the selected partitions; perform image comparison on the stored baseline image regions and the extracted image instance regions; and send an alert that the website has been defaced when a result of the image comparison exceeds a first threshold.

[0018] Preferably, each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a image intensity value, and the processor is further programmed to calculate a first image intensity value for each of the stored baseline image regions by totaling up the image intensity values of the pixels within each of the stored baseline image regions; calculate a second image intensity value for each of the image instance regions by totaling up the image intensity values of the pixels within each of the extracted image instance regions, and wherein the result of the image comparison is dependent on the difference between the first image intensity value and the second image intensity value.

[0019] Preferably, each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a red channel value, a green channel value and a blue channel value, and the processor is further programmed to calculate a first red channel value for each of the stored baseline image regions by totaling up the red channel values of each pixel within each of the stored baseline image regions and calculate a second red channel value for each of the image instance regions by totaling up the red channel values of each pixel within each of the extracted image instance regions. The processor is further programmed to calculate a first green channel value for each of the stored baseline image regions by totaling up the green channel values of each pixel within each of the stored baseline image regions; calculate a second green channel value for each of the image instance regions by totaling up the green channel values of each pixel within each of the extracted

image instance regions; calculate a first blue channel value for each of the stored baseline image regions by totaling up the red channel values of each pixel within each of the stored baseline image regions, calculate a second blue channel value for each of the image instance regions by totaling up the red channel values of each pixel within each of the extracted image instance regions, and wherein the result of the image comparison is dependent on the difference between the first red channel value and the second red channel value, and on the difference between the first green channel value and the second green channel value, and on the difference between the first blue channel value and the second blue channel value.

[0020] Preferably, each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a image intensity value, a red channel value, a green channel value and a blue channel value, and the processor is further programmed to compare the image intensity values of each pixel of the stored baseline image regions with the image intensity values of each pixel of the extracted image instance regions to determine a number of pixels whose image intensity values has changed and compare the red channel values of each pixel of the stored baseline image regions with the image intensity values of each pixel of the extracted image instance regions to determine a number of pixels whose red channel values has changed. The processor is further programmed to compare the green channel values of each pixel of the stored baseline image regions with the image intensity values of each pixel of the extracted image instance regions to determine a number of pixels whose green channel values has changed; compare the blue channel values of each pixel of the stored baseline image regions with the image intensity values of each pixel of the extracted image instance regions to determine a number of pixels whose blue channel values has changed, and wherein the result of the image comparison is dependent on the number of pixels whose image intensity value has changed, and on the number of pixels whose red channel value has changed, and on the number of pixels whose green channel value has changed and on the number of pixels whose blue channel value has changed.

[0021] Preferably, the processor is further programmed to obtain a baseline HTML content of the website; store the baseline HTML content in the database; obtain a HTML content instance of the website at the polled interval; perform content comparison on the stored baseline HTML content and the HTML content instance; and send an alert that the website has been defaced when a result of the content comparison exceeds a second threshold.

[0022] Preferably, the processor is further programmed to count a number of links in the stored baseline HTML content and the HTML content instance; count a number of scripts in the stored baseline HTML content and the HTML content instance; and count a number of images in the stored baseline HTML content and the HTML content instance.

[0023] Preferably, the processor is further programmed to perform a first integrity comparison on the baseline image and the image instance; perform a second integrity comparison on the baseline HTML content and the HTML content instance; and send an alert that the website has been defaced when a result of the first integrity comparison and a result of the second integrity comparison exceeds a third threshold.

[0024] Preferably, the processor is further programmed to hash a image in the baseline image to obtain a first hash value; hash a image in the image instance to obtain the second hash value; and compare the first hash value and the second hash value.

[0025] Preferably, the processor is further programmed to hash a script in the baseline HTML content to obtain a first hash value; hash a script in the HTML content instance to obtain the second hash value; and compare the first hash value and the second hash value.

[0026] Preferably, the processor is further programmed to check the HTML content instance for malwares and send an alert that the website has been defaced when at least one malware is detected.

[0027] Preferably, the processor is further programmed to wait for a predetermined period to lapse after obtaining the baseline HTML content of the website; obtain another baseline HTML content of the website; compare the baseline HTML content with the another baseline HTML content and allow the second threshold and third threshold to be adjusted based on this comparison.

[0028] Preferably, the partitioning algorithm is anyone of the following: a 4 by 4 grid; a 3 by 3 grid, a 5 by 5 grid and a 6 by 6 grid.

[0029] The invention will now be described in detail with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0030] The accompanying figures illustrate disclosed embodiment(s) and serve to explain principles of the disclosed embodiment(s). It is to be understood, however, that these drawings are presented for purposes of illustration only, and not for defining limits of the application.

[0031] Figure 1 shows a schematic diagram of a website defacement monitoring system accordingly to an exemplary embodiment.

[0032] Figure 2 shows a flow chart for a website defacement monitoring system performing image comparison.

[0033] Figure 3 shows a baseline image being partitioned into a 4 by 4 grid format.

[0034] Figure 4 shows a flow chart for a website defacement monitoring system performing content comparison.

[0035] Figure 5 shows a flow chart for a website defacement monitoring system performing integrity comparison.

[0036] Figure 6 shows a flow chart for a website defacement monitoring system performing malware checks.

[0037] Figure 7 shows a flow chart for obtaining baseline HTML content multiple times or multiple baselining.

[0038] Figure 8 shows a table with a threshold value being adjusted according to the results of multiple baselining.

[0039] Exemplary, non-limiting embodiments of the present application will now be described with references to the above-mentioned figures.

DETAILED DESCRIPTION

[0040] Figure 1 shows a system for monitoring website defacements. The system comprises core engine 101, database 102 and image comparison engine 103. Core engine 101 has web module 104 that is responsible for retrieving content from the website. This content can be an image capture of the website. This content can also be the HTML (HyperText Markup Language) content of the website. Core engine 101 also has analytical module 105 that is responsible for determining if the website has been defaced. Analytical module 105 can make this determination by retrieving data from image comparison engine 103. Database 102 is for storing data and this stored data can be accessed or retrieved by image comparison engine 103. Content comparison engine 106, integrity comparison engine 107 and signature comparison engine 108 will be described later.

[0041] Figure 2 describes a preferred embodiment of the invention for monitoring website defacement using image comparison engine 103. In step 201, web module 104 of core engine 101 obtains the baseline image of a website by screen-capturing the website. This can be done by rendering the screen capture to a Portable Network Graphic (PNG) image with the Webkit layout engine that powers Safari and Chrome browsers. Alternatively, the baseline image can be saved in other media formats like JPEG (Joint Photographic Experts Group) etc. Optionally, the baseline image may be further edited, resized, cropped. This can be done by using the ImageMagick library for PHP (Hypertext Preprocessor). The purpose of obtaining the baseline image is for it to act as a reference to which subsequent screen captures would be compared against.

[0042] In step 202, core engine 101 partitions the baseline image into a plurality of smaller baseline image regions according to the partitions in a partitioning algorithm. The portion of the baseline image within a partition is the baseline image region. An example of a partitioning algorithm can be one that partitions the baseline image into a 4 by 4 grid format, resulting in 16 partitions. These partitions can be of the same size. Each partition in the partitioning algorithm can have an index value, so that the partitions can be uniquely identified. Example of index values for partitions 302 of baseline image 301 in a 4 by 4 grid format can be found in Figure 3 i.e. A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4, D1, D2, D3 and D4. The partitioning algorithm can also be a 3 by 3, 5 by 5, 3 by 4, 4 by 5, 5 by 5, 6 by 6 grid format, and any other conceivable format.

[0043] In step 203 of figure 2, the user is allowed to select from the partitions. The partitions are presented in a user interface to the user. The user interface can be rendered by a front-end web engine. The user can select the partitions by determining which baseline image regions (which are within the partitions) should be used for the image comparison. The advantage is that the user can specify exactly the baseline image regions to be used for the image comparison, and therefore optionally exclude baseline image regions with dynamic content from the image comparison. Dynamic content constantly change and therefore excluding baseline image regions that have dynamic content from the image comparison helps to reduce the amount of false positives (false alarm) when monitoring websites.

[0044] It is immaterial how many partitions the user chooses. The user can choose one partition, and even all the partitions. The number of partitions chosen by the user has no bearing on the effectiveness of the invention. The crucial factor is that the user knows his website best, and therefore would be the best person to select which baseline image regions should be used or is suitable for the purposes of the image comparison.

[0045] Once the user has selected the partitions, in step 204, the baseline image regions which correspond to the selected partitions are stored in database 102.

[0046] Monitoring of the website occurs at a predefined polling interval. At the start of a polling interval, an image instance of the website is obtained (step 205). This is done by web module 104 of core engine 101 screen-capturing the website at that particular point in time (or the start of a polled interval). The screen-capturing is done by rendering the screen capture to a PNG file or JPEG file (similar to how it is done for the baseline image). This PNG file or JPEG file is the image instance.

[0047] At step 206, the baseline image regions which correspond to the selected partitions, are retrieved from the database.

[0048] At step 207, the image instance is partitioned into a plurality of image instance regions according to the partitions in the partitioning algorithm. This partitioning algorithm is the same partitioning algorithm used in step 202. For instance, if the partitioning algorithm used in step 202 was a 4 by 4 grid format, the partitioning algorithm used here would also be a

4 by 4 grid format. This is to ensure that in the upcoming steps, the baseline image regions and image instance regions within the same partitions are being compared.

[0049] At step 208, the image instance regions that correspond to the selected partitions (selected in step 203) are extracted from the image instance. Core engine 101 can identify these selected partitions by their index values. This is to ensure that in the upcoming steps, the baseline image regions and image instance regions within the same selected partitions are being compared.

[0050] At step 209, image comparison is performed between the baseline image regions and the image instance regions. Both these baseline image regions and image instance regions correspond to the selected partitions. Further, the image comparison is done between the baseline image region and image instance region that corresponds to the same selected partition. For example, image comparison is done between baseline image region that corresponds to selected partition with index value “A1” and image instance region that corresponds to selected partition with index value “A1”. In other words, comparison is always done between a baseline image region and its corresponding image instance region. This is to ensure that the baseline image regions and image instance regions within the same selected partitions are being compared.

[0051] The image comparison is done by image comparison engine 103. Based on the results of the image comparison done by image comparison engine 103, in step 210, analytical module 105 of core engine 101 determines if the threshold value has been reached or exceeded. If the threshold value has been reached or exceeded, a notification module will be triggered (the notification module will be explained later) to alert the user that the website has been defaced. If the threshold value has not been reached or exceeded, the notification module will not be triggered, as analytical module 105 of core engine 101 has determined that the website has not been defaced. Instead, web module 104 of core engine 101 will wait for the start of the next polling interval to repeat steps 205 to 210 to continue the monitoring process.

[0052] In a preferred embodiment, the image comparison done by image comparison engine 103 is a comparison on the image intensities of the baseline image region and the image instance region. The baseline image region will comprise a plurality of pixels and the image instance region will comprise a plurality of pixels. Each of these pixels has a pixel

value which describes how bright a pixel is, and what color the pixel should be. For grayscale images, the pixel value is a single number which represents the brightness of the pixel. The most common pixel format is the byte image, where this number is stored as an 8-bit integer giving a range of possible values from 0 to 255, 0 indicating black, and 255 indicating white, with values in-between making up the different shades of grey.

[0053] The image intensity of the baseline image region is the sum of the brightness pixel values of all the pixels within the baseline image region. The image intensity of the image instance region is the sum of the brightness pixel values of all the pixels within the image instance region. A delta value is determined by calculating the difference between the image intensity of the baseline image region and the image intensity of the corresponding image instance region. A delta value is calculated for each baseline image region and corresponding image instance region combination.

[0054] Image comparison engine 103 then passes the delta values to analytical module 105 of core engine 101. If any one of the delta value is equal or larger than a threshold value, the notification module can be triggered indicating the website has been defaced. Alternatively, only if all the delta values are equal or larger than a threshold value, would the notification module is triggered. In other words, the triggering step can be customized to suit the nature of the website. If the website is highly dynamic, it may require a large percentage or all of the delta values to exceed the threshold before the notification module is triggered. This therefore helps to reduce the amount of false positives (false alarm) when monitoring websites having dynamic content.

[0055] In another preferred embodiment, the image comparison done by image comparison engine 103 is a comparison on the channel values of the baseline image regions and the image instance regions. A color perceived by the human eye can be defined by a linear combination of the three primary colors red, green and blue. These three colors form the basis of the RGB color-space. An RGB image has three channels: red, green and blue. If the RGB image is 24-bit, each channel has 8 bits, for red, green and blue. In other words, the image is composed of three images (one for each channel), where each image can store discrete pixels with conventional brightness intensities between 0 and 255. If the RGB image is 48-bit (very high resolution), each channel is made of 16 bit images.

[0056] The channel value (red) of the baseline image region is calculated by summing up the channel values (red) of the pixels in the baseline image region. The channel value (red) of the corresponding image instance region is calculated by summing up the channel values (red) of the pixels in the image instance region. A delta value (red) is determined by calculating the difference between the channel value (red) of the baseline image region and the channel value (red) of the corresponding image instance region..

[0057] The channel value (green) of the baseline image region is calculated by summing up the channel values (green) of the pixels in the baseline image region. The channel value (green) of the corresponding image instance region is calculated by summing up the channel values (green) of the pixels in the image instance region. A delta value (green) is determined by calculating the difference between the channel value (green) of the baseline image region and the channel value (green) of the corresponding image instance region..

[0058] The channel value (blue) of the baseline image region is calculated by summing up the channel values (blue) of the pixels in the baseline image region. The channel value (blue) of the corresponding image instance region is calculated by summing up the channel values (blue) of the pixels in the image instance region. A delta value (blue) is determined by calculating the difference between the channel value (blue) of the baseline image region and the channel value (blue) of the corresponding image instance region. A delta value (red), a delta value (green) and a delta value (blue) is calculated for each baseline image region and corresponding image instance region combination, and image comparison engine 103 then passes these values to analytical module 105 of core engine 101.

[0059] If anyone of delta value (red), delta value (green) and delta value (blue) is not equal or larger than a threshold value, the notification module can be triggered indicating the website has been defaced. Alternatively, only if all of delta value (red), delta value (green) and delta value (blue) is equal or larger than a threshold value, would the notification module is triggered. In other words, the triggering step can be customized to suit the nature of the website. If the website is highly dynamic, it may require a large percentage or all of the delta values to exceed the threshold before the notification module be triggered. This therefore helps to reduce the amount of false positives (false alarm) when monitoring websites having dynamic content.

[0060] In another preferred embodiment, the image comparison done by image comparison engine 103 is a comparison of each pixel in the baseline image region with each pixel in the corresponding image instance region. Each pixel value (brightness value, channel value (red), channel value (green), channel value (blue)) of each pixel in the baseline image region is compared to each pixel value (brightness value, channel value (red), channel value (green), channel value (blue)) of each pixel in the corresponding image instance region. Each pixel value that is different would result in a counter value incrementing by 1. For example, if between a baseline image region (having 500 pixels, and therefore 2000 pixel values) and a image instance region (which therefore also has 500 pixels and 2000 pixel values), 10 pixel values are different, the counter value would be 10. This comparison by pixel is done for each baseline image region and corresponding image instance region combination.

[0061] Image comparison engine 103 then passes the counter value to analytical module 105 of core engine 101. If the counter value is equal or larger than a threshold value, the notification module will be triggered indicating the website has been defaced.

[0062] In another preferred embodiment, the image comparison done by image comparison engine 103 is a comparison on the hash values of a baseline image region and the corresponding image instance region. The binary code representation of a baseline image region is passed into a hashing algorithm which will output a hash value. Examples of hashing algorithms are Secure Hash Algorithm (SHA) and MD5 etc. The binary code representation of the corresponding image instance region is passed into the hashing algorithm which will output a hash value. These two hash values are then compared. This comparison of hash values is done for each baseline image region and corresponding image instance region combination, and the results are passed to analytical module 105 of core engine 101.

[0063] If at least one baseline image region and corresponding image instance region combination has hash values that are different, the notification module can be triggered indicating the website has been defaced. Alternatively, only if all the baseline image region and corresponding image instance region combinations have hash values that are different, would the notification module be triggered. In other words, the triggering step can be customized to suit the nature of the website. This therefore helps to reduce the amount of false positives (false alarm) when monitoring websites having dynamic content.

[0064] In another preferred embodiment, the image comparison done by image comparison engine 103 can be any combination (with any sequence) of the above four described methods i.e. comparison by image intensity, comparison by channel value and comparison of each pixel, comparison of hash values.

[0065] The notification module when triggered by core engine 101 can send an alert to the user. The alert can be in any form through any communication medium e.g. email, Short Message Service (SMS) and the like. For Security Operating Centres (SOC) operations, Simple Network Management Protocol (SNMP) and syslog can also be additionally included. A reporting module can be triggered to provide both historical and ad-hoc reports to the users via the front-end web user interface. Customized reports for SOC operation, with co-branding can also be supported.

[0066] In a further embodiment, the system further comprises content comparison engine 106. Content comparison engine 106 is for comparing the content of the baseline HTML content and the HTML content instance.

[0067] Figure 4 describes an embodiment of the website defacement monitoring system using content comparison engine 106. In step 401, web module 104 of core engine 101 obtains the baseline HTML content of the website. The baseline HTML content is obtained by downloading it from the website's web server over the internet via the HTTP protocol. The purpose of obtaining the baseline HTML content is for it to act as a reference to which subsequent HTML content instances or HTML content captures of the website would be compared against.

[0068] In step 402, the baseline HTML content is stored in database 102.

[0069] Monitoring of the website occurs at a predefined polling interval. At the start of a polling interval, a HTML content instance of the website is obtained (step 403). The HTML content instance is obtained by downloading it from the website's web server over the internet via the HTTP protocol.

[0070] At step 404, the baseline HTML content is retrieved from the database.

[0071] At step 405, the content of the baseline HTML content and the HTML content instance are compared. The content comparison can include, though limited to, the number of Universal Resource Locator (URL) links; the HTTP header (such as the title); the text; the number of scripts; the number of images; the number of various HTML tags, the number of iframes, the number of style sheets and the like. For example, if the number of URL links of the baseline HTML content and the HTML content instance are different, this is perhaps an indication that the website has been defaced. All kinds of HTML tags, embedded objects and images can be counted and compared.

[0072] Certain pre-defined keywords can also be searched for in the HTML content instance. For example, the baseline HTML content has the corporate name of a company occurring many times. If the HTML content instance does not have the corporate name of the company occurring, this is perhaps an indication that the website has been defaced. Comparing by pre-defined keywords allow for easy monitoring as content comparison engine 106 simply needs to check if these keywords are present. If these keywords are absent, the monitoring system may trigger an alert to say that the webpage has been defaced.

[0073] The results of the content comparison of the baseline HTML content and the HTML content instance by content comparison engine 106 are then sent to analytical module 105 of core engine 101 to determine if a threshold value has been met at step 406. If the threshold value has been met, the notification module will be triggered to alert the user that the website has been defaced.

[0074] If the threshold value has not been met, the notification module will not be triggered. Analytical module 105 of core engine 101 has determined that the website has not been defaced. Instead, web module 104 of core engine 101 will wait for the start of the next polling interval to repeat steps 403 to 406 to continue the monitoring process.

[0075] In a further embodiment, the system further comprises integrity comparison engine 107. Integrity comparison engine 107 checks the integrity of the contents (for example images, scripts, URL links) of the website.

[0076] Figure 5 describes an embodiment of the website defacement monitoring system using integrity comparison engine 107. In step 501, web module 104 of core engine

101 obtains the baseline image and baseline HTML content of the website. The baseline image is obtained in a manner similar to what was described for step 201. The baseline HTML content is obtained in a manner similar to what was described for step 401.

[0077] In step 502, the baseline image and baseline HTML content are stored in database 102.

[0078] Monitoring of the website occurs at a predefined polling interval. At the start of a polling interval, an image instance and HTML content instance of the website is obtained (step 503). The image instance of the website is obtained in a manner similar to what was described for step 205. The HTML content instance is obtained in a manner similar to what was described for step 403.

[0079] At step 504, baseline image and baseline HTML content are retrieved from the database.

[0080] At step 505, integrity comparison is performed on the baseline image and the image instance. One example of integrity comparison is to compare the hash value of a particular image in the baseline image and the hash value of the same image in the image instance. The hashing is done by passing the binary code representation of the image into a hashing algorithm. Different hash values may indicate that the website has been defaced.

[0081] At step 506, integrity comparison is performed on the baseline HTML content and the HTML content instance. One example of integrity comparison is to compare the hash value of a particular script in the baseline HTML content and the hash value of the same script in the HTML content instance. The hashing is done by passing the HTML code of the script into a hashing algorithm. Different hash values may indicate that the website has been defaced.

[0082] Based on the results of the integrity comparison by integrity comparison engine 107, analytical module 105 of core engine 101 determines if the threshold value has been met at step 507. If the threshold value has been met, the notification module will be triggered to alert the user that the website has been defaced.

[0083] If the threshold value has not been met, the notification module will not be triggered. Analytical module 105 of core engine 101 has determined that the website has not

been defaced. Instead, web module 104 of core engine 101 will wait for the start of the next polling interval to repeat steps 503 to 507 to continue the monitoring process.

[0084] In a further embodiment, the system further comprises signature comparison engine 108. Signature comparison engine 108 checks the website for malwares (malicious software). Signature comparison engine 108 does this by comparing the binary files that the website is hosting against known virus signatures, and web objects like java applet and image objects. Malwares can also be in the form of scripts instead of a binary file e.g. JavaScripts. Signature comparison engine 108 can check the website for known malicious JavaScripts within the HTML and JavaScript file.

[0085] Figure 6 describes an embodiment of the website defacement monitoring system using signature comparison engine 108. Monitoring of the website occurs at a predefined polling interval. At the start of a polling interval, the HTML content instance of the website is obtained (step 601). The HTML content instance is obtained by downloading it from the website's web server over the internet via the HTTP protocol.

[0086] At step 602, the HTML content instance is checked for traces of malwares. This is done by signature comparison engine 108. Signature comparison engine 108 checks the HTML content instance for known malicious JavaScripts. Signature comparison engine 108 also checks for known virus signatures, and web objects like java applet and image objects. The result of the malware check is then sent to analytical module 105 of core engine 101.

[0087] If a virus or malware has been detected, the notification module will be triggered to notify the user that the website has been defaced. If no virus or malware has been detected, the notification module will not be triggered and web module 104 of core engine 101 will wait for the start of the next polling interval to repeat steps 601 and 602 to continue the monitoring process

[0088] In steps 401 and 501, the baseline HTML content is obtained only once. In a further embodiment, the baseline HTML content is obtained a plurality of times at a polled interval as described in Figure 7. Referring to step 701 of figure 7, web module 104 of core engine 101 obtains a first baseline HTML content of the website by downloading it from the

website's web server over the internet via the HTTP protocol. In step 702, web module 104 of core engine 101 waits for a polled interval to expire, before obtaining a second baseline HTML content of the website in step 703. In step 704, web module 104 of core engine 101 waits for another polled interval to expire, before obtaining a third baseline HTML content of the website in step 705. This is known as multiple baselining. In step 706, the first, second and third baseline HTML contents are then compared. The comparison includes but is not limited to, scripts, HREF title tags and Cascading Style Sheets (CSS) and the like.

[0089] The advantage of performing multiple baselining is that the threshold values used by analytical module 105 of core engine 101 can be adjusted to more accurately reflect the dynamic and static portions of the website. For example, in figure 8, multiple baselining has determined that the number of images on the website fluctuates between a lower value of 10 images and an upper value of 20 images. This determination can then be used as a guide to set the threshold value. In this example, the threshold value is set to ± 5 as the number of images on the website fluctuates between 10 and 20 images. Therefore, if the number of images in the baseline HTML content is 15, while the number of images in the HTML content instance is 10, this would be within the threshold value and no alert would be raised.

[0090] Further, multiple baselining can also help determine the ideal values for the baseline HTML content (for which subsequent HTML content instances will be compared against). In other words, the baseline HTML content is not simply an initial HTML content capture of the website, but it is further tweaked and adjusted based on the values obtained from the multiple baselining. The values of the baseline HTML content may be the average of the values obtained by the multiple baselining. In the example as provided in figure 8, the ideal number of images in the baseline HTML content is the average of the lower value of 10 images and the upper value of 20 images i.e. 15 images. This ideal value is possible through the multiple baselining. This ideal value is merely a suggested value to the user. The user can further adjust these values of the baseline HTML content using the user interface provided by the front-end web engine.

[0091] Although figure 7 describes that the baseline HTML content is obtained three times, it would be obvious to a skilled person that the number of times does not quite matter, as long as it is multiple times (or more than one). Obviously, the more times the baseline HTML content is obtained, the more accurately the system can determine which portions are

static and which portions are dynamic. Furthermore, the length of the polled interval does not quite matter. Obviously, the longer the polled interval, the more accurately the system can determine which portions are static and which portions are dynamic, as the website may not have been updated during a short polled interval. However, the trade-offs would be that the multiple baselining process would take longer.

[0092] In another preferred embodiment, the website defacement monitoring system can have any combination of the image comparison engine 103, content comparison engine 106, integrity comparison engine 107 and signature comparison engine 108. Although in these specifications, the image comparison is described first, followed by the content comparison, integrity comparison and then the signature comparison, one skilled in the art can appreciate that it does not quite matter which form of comparison is employed first.

[0093] One single server can house core engine 101, image comparison engine 103, content comparison engine 106, integrity comparison engine 107, signature comparison engine 108 and database 102. This server can have one or more processors for implementing core engine 101, image comparison engine 103, content comparison engine 106, integrity comparison engine 107 and signature comparison engine 108. Alternatively, core engine 101, image comparison engine 103, content comparison engine 106, integrity comparison engine 107, signature comparison engine 108 and database 102 can be in different servers. One skilled in the art will appreciate that other variations are possible.

[0094] In the application, unless specified otherwise, the terms "comprising", "comprise", and grammatical variants thereof, intended to represent "open" or "inclusive" language such that they include recited elements but also permit inclusion of additional, non-explicitly recited elements.

[0095] It will be apparent that various other modifications and adaptations of the application will be apparent to the person skilled in the art after reading the foregoing disclosure without departing from the spirit and scope of the application and it is intended that all such modifications and adaptations come within the scope of the appended claims.

CLAIMS

1. A method for monitoring a website for defacement comprising the steps of :-
 - obtaining a baseline image of the website;
 - partitioning the baseline image into a plurality of baseline image regions according to a plurality of partitions in a partitioning algorithm;
 - allowing the partitions to be selected by a user by presenting the partitions in a user interface to the user;
 - obtaining the selected partitions from the user via the user interface;
 - storing the baseline image regions which correspond to the selected partitions in a database;
 - obtaining a image instance of the website at a polled interval;
 - partitioning the image instance into a plurality of image instance regions according to the plurality of partitions in the partitioning algorithm;
 - extracting the image instance regions which correspond to the selected partitions;
 - performing image comparison on the stored baseline image regions and the extracted image instance regions; and
 - sending an alert that the website has been defaced when a result of the image comparison exceeds a first threshold.
2. The method of claim 1 wherein each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a image intensity value, and wherein the step of performing image comparison on the stored baseline image regions and the extracted image instance regions comprises the steps of :-
 - calculating a first image intensity value for each of the stored baseline image regions by totaling up the image intensity values of the pixels;
 - calculating a second image intensity value for each of the extracted image instance regions by totaling up the image intensity values of the pixels; and
 - wherein the result of the image comparison is dependent on the difference between the first image intensity value and the second image intensity value.

3. The method of claim 1 wherein each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a red channel value, a green channel value and a blue channel value, and wherein the step of performing image comparison on the stored baseline image regions and the extracted image instance regions comprises the steps of :-

- calculating a first red channel value for each of the stored baseline image regions by totaling up the red channel values of each pixel;

- calculating a second red channel value for each of the extracted image instance regions by totaling up the red channel values of each pixel;

- calculating a first green channel value for each of the stored baseline image regions by totaling up the green channel values of each pixel;

- calculating a second green channel value for each of the extracted image instance regions by totaling up the green channel values of each pixel;

- calculating a first blue channel value for each of the stored baseline image regions by totaling up the blue channel values of each pixel;

- calculating a second blue channel value for each of the extracted image instance regions by totaling up the blue channel values of each pixel; and

- wherein the result of the image comparison is dependent on the difference between the first red channel value and the second red channel value, and on the difference between the first green channel value and the second green channel value, and on the difference between the first blue channel value and the second blue channel value.

4. The method of claim 1 wherein each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a image intensity value, a red channel value, a green channel value and a blue channel value, and wherein the step of performing image comparison on the stored baseline image regions and the extracted image instance regions comprises the steps of :-

- comparing the image intensity value of each pixel of the stored baseline image regions with the image intensity value of each pixel of the extracted image instance regions to determine a number of pixels whose image intensity value has changed;

- comparing the red channel value of each pixel of the stored baseline image regions with the red channel value of each pixel of the extracted image instance regions to determine a number of pixels whose red channel value has changed;

comparing the green channel value of each pixel of the stored baseline image regions with the green channel value of each pixel of the extracted image instance regions to determine a number of pixels whose green channel value has changed;

comparing the blue channel value of each pixel of the stored baseline image regions with the blue channel value of each pixel of the extracted image instance regions to determine a number of pixels whose blue channel value has changed; and

wherein the result of the image comparison is dependent on the number of pixels whose image intensity value has changed, and on the number of pixels whose red channel value has changed, and on the number of pixels whose green channel value has changed and on the number of pixels whose blue channel value has changed.

5. The method of any one of the preceding claims further comprising the steps of :-

obtaining a baseline HTML content of the website;

storing the baseline HTML content in the database;

obtaining a HTML content instance of the website at the polled interval;

performing content comparison on the stored baseline HTML content and the HTML content instance; and

sending an alert that the website has been defaced when a result of the content comparison exceeds a second threshold.

6. The method of claim 5 where the step of performing content comparison on the stored baseline HTML content and the HTML content instance comprises at least one of the steps of :-

counting a number of links in the stored baseline HTML content and the HTML content instance;

counting a number of scripts in the stored baseline HTML content and the HTML content instance; and

counting a number of images in the stored baseline HTML content and the HTML content instance.

7. The method of claim 5 or 6 further comprising :-

performing a first integrity comparison on the baseline image and the image instance;

performing a second integrity comparison on the baseline HTML content and the HTML content instance; and

sending an alert that the website has been defaced when a result of the first integrity comparison and a result of the second integrity comparison exceeds a third threshold.

8. The method of claim 7 wherein the step of performing a first integrity comparison on the baseline image and the image instance comprises the steps of :-

- hashing a image in the baseline image to obtain a first hash value;
- hashing a image in the image instance to obtain a second hash value; and
- comparing the first hash value and the second hash value.

9. The method of claim 7 wherein the step of performing a second integrity comparison on the baseline HTML content and the HTML content instance comprises the steps of :-

- hashing a script in the baseline HTML content to obtain a first hash value;
- hashing a script in the HTML content instance to obtain a second hash value; and
- comparing the first hash value and the second hash value.

10. The method of any one of claims 5 to 9 further comprising the steps of checking the HTML content instance for malwares and sending an alert that the website has been defaced when at least one malware is detected.

11. The method of any one of claims 5 to 10 further comprising the steps of :-

- waiting for a predetermined period to lapse after obtaining the baseline HTML content of the website;

- obtaining another baseline HTML content of the website;

- comparing the baseline HTML content with the another baseline HTML content and allowing the second threshold and third threshold to be adjusted based on this comparison.

12. The method of any one of the preceding claims wherein the partitioning algorithm is anyone of the following :- a 4 by 4 grid; a 3 by 3 grid, a 5 by 5 grid and a 6 by 6 grid.

13. A system for monitoring a website for defacement comprising a database and at least one processor programmed to :-

- obtain a baseline image of the website;

- partition the baseline image into a plurality of baseline image regions according to a plurality of partitions in a partitioning algorithm;

allow the partitions to be selected by a user by presenting the partitions in a user interface to the user;
obtain the selected partitions from the user via the user interface;
store the baseline image regions which correspond to the selected partitions in the database;
obtain a image instance of the website at a polled interval;
partition the image instance into a plurality of image instance regions according to the partitions;
extract the image instance regions which correspond to the selected partitions;
perform image comparison on the stored baseline image regions and the extracted image instance regions; and
send an alert that the website has been defaced when a result of the image comparison exceeds a first threshold.

14. The system of claim 13 wherein each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a image intensity value, and wherein the at least one processor is further programmed to :-

calculate a first image intensity value for each of the stored baseline image regions by totaling up the image intensity values of the pixels within each of the stored baseline image regions;

calculate a second image intensity value for each of the image instance regions by totaling up the image intensity values of the pixels within each of the extracted image instance regions; and

wherein the result of the image comparison is dependent on the difference between the first image intensity value and the second image intensity value

15. The system of claim 13 wherein each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a red channel value, a green channel value and a blue channel value, and wherein the at least one processor is further programmed to :-

calculate a first red channel value for each of the stored baseline image regions by totaling up the red channel values of each pixel within each of the stored baseline image regions;

calculate a second red channel value for each of the image instance regions by totaling up the red channel values of each pixel within each of the extracted image instance regions;

calculate a first green channel value for each of the stored baseline image regions by totaling up the green channel values of each pixel within each of the stored baseline image regions;

calculate a second green channel value for each of the image instance regions by totaling up the green channel values of each pixel within each of the extracted image instance regions;

calculate a first blue channel value for each of the stored baseline image regions by totaling up the red channel values of each pixel within each of the stored baseline image regions;

calculate a second blue channel value for each of the image instance regions by totaling up the red channel values of each pixel within each of the extracted image instance regions; and

wherein the result of the image comparison is dependent on the difference between the first red channel value and the second red channel value, and on the difference between the first green channel value and the second green channel value, and on the difference between the first blue channel value and the second blue channel value.

16. The system of claim 13 wherein each stored baseline image regions and extracted image instance regions comprises pixels, each pixel having a image intensity value, a red channel value, a green channel value and a blue channel value, and wherein the at least one processor is further programmed to :-

compare the image intensity values of each pixel of the stored baseline image regions with the image intensity values of each pixel of the extracted image instance regions to determine a number of pixels whose image intensity values has changed;

compare the red channel values of each pixel of the stored baseline image regions with the image intensity values of each pixel of the extracted image instance regions to determine a number of pixels whose red channel values has changed;

compare the green channel values of each pixel of the stored baseline image regions with the image intensity values of each pixel of the extracted image instance regions to determine a number of pixels whose green channel values has changed;

compare the blue channel values of each pixel of the stored baseline image regions with the image intensity values of each pixel of the extracted image instance regions to determine a number of pixels whose blue channel values has changed; and

wherein the result of the image comparison is dependent on the number of pixels whose image intensity value has changed, and on the number of pixels whose red channel value has changed, and on the number of pixels whose green channel value has changed and on the number of pixels whose blue channel value has changed.

17. The system of anyone of claims 13 to 16 wherein the at least one processor is further programmed to:-

obtain a baseline HTML content of the website;

store the baseline HTML content in the database;

obtain a HTML content instance of the website at the polled interval;

perform content comparison on the stored baseline HTML content and the HTML content instance; and

send an alert that the website has been defaced when a result of the content comparison exceeds a second threshold.

18. The system of claim 17 wherein the at least one processor is further programmed to:-

count a number of links in the stored baseline HTML content and the HTML content instance;

count a number of scripts in the stored baseline HTML content and the HTML content instance; and

count a number of images in the stored baseline HTML content and the HTML content instance.

19. The system of claim 17 or 18 wherein the at least one processor is further programmed to:-

perform a first integrity comparison on the baseline image and the image instance;

perform a second integrity comparison on the baseline HTML content and the HTML content instance; and

send an alert that the website has been defaced when a result of the first integrity comparison and a result of the second integrity comparison exceeds a third threshold.

20. The system of claim 19 wherein the at least one processor is further programmed to:-
hash a image in the baseline image to obtain a first hash value;
hash a image in the image instance to obtain a second hash value; and
compare the first hash value and the second hash value.
21. The system of claim 19 wherein the at least one processor is further programmed to:-
hash a script in the baseline HTML content to obtain a first hash value;
hash a script in the HTML content instance to obtain a second hash value; and
compare the first hash value and the second hash value.
22. The system of any one of claims 17 to 21 wherein the at least one processor is further programmed to check the HTML content instance for malwares and send an alert that the website has been defaced when at least one malware is detected.
23. The system of any one of claims 17 to 22 wherein the at least one processor is further programmed to:-
wait for a predetermined period to lapse after obtaining the baseline HTML content of the website;
obtain another baseline HTML content of the website;
compare the baseline HTML content with the another baseline HTML content and
allow the second threshold and third threshold to be adjusted based on this comparison.
24. The system of any one claims 13 to 23 wherein the partitioning algorithm is anyone of the following :- a 4 by 4 grid; a 3 by 3 grid, a 5 by 5 grid and a 6 by 6 grid.