

# A CONCEPTUAL APPROACH TO DETECT WEBDEFACEMENT THROUGH ARTIFICIAL INTELLIGENCE

Ebot Ebot Enaw, Djoursoubo Pagou Prosper  
University of Yaounde I, Cameroon  
National Advanced School of Engineering  
Department of Computer Sciences

## Abstract

Over the past couple of years, the number of webdefacement incidents has considerably increased making it one of the primary concerns in cybersecurity.

Many approaches have been developed to automate the detection of these attacks, however they are not quite efficient as they generated too many false-positives.

Therefore, in a bid to develop an efficient system to detect webdefacements while minimizing false positives, we designed an approach based on artificial intelligence's concepts like anomaly detection, inference and machine learning. Our article is structured as follows: section 1 introduces the article, section 2 presents some research papers related to our topic, section 3 states the problem, section 4 presents webdefacement attacks, section 5 presents the main anomaly detection techniques, section 6 presents our solution.

**Keywords:** *web defacement, anomaly detection, machine learning, expert system.*

## 1 Introduction

With the rapid growth of the Internet, websites have become very important for businesses, government and even for individuals since they serve multiple purposes including: interaction with customers, presentation of products/services, and dissemination of information. However, because of their ever increasing importance, websites have also become one of the primary targets of hackers leading to the advent of a new type of attack called webdefacement.

Webdefacement consists of fraudulently altering the content of a website. Usually this unauthorized alteration is done on the home page of the target website and is aimed at disseminating propaganda or trick users into downloading malicious contents.

Though some techniques have been developed to automatically detect webdefacements namely page hashing or page DOM analysis, these techniques are not efficient as they generated too many false positives.

Therefore new approaches need to be developed to detect webdefacements while minimizing false positives especially in the present context where web pages are regularly and dynamically changed.

These approaches might definitely need intelligent algorithms that can adapt to different situations, infer on data and learn from past experiences.

The aim of this paper is to present a practical approach to detect webdefacement by using artificial intelligence concepts including anomaly detection, inferences, machine learning to reduce false-positives.

## 2 Related work

Some research has been done on topics related to this issue namely [4] that proposes a solution to prevent webdefacement based on read-only storage media. Their approach focuses on improving integrity by ensuring that neither web content nor system configuration can be altered on the web server. Their solution consists of building a live CD using the *catalyst* platform that contains the web content and the system packages. Since for normal operations, operating systems need certain parts of file system to be writable especially for storing temporary file, name pipes and socket, they used *zisosfs* which is a compressed, read-only file system which dynamically decompresses each file into RAM on access at runtime, circumventing the need to keep the whole decompressed file system in memory. In order to avoid breaching the server through a RAM attack, they proposed to restart the server frequently in order to clear the RAM content; and to avoid the downtime induced by the restarting of the server they proposed an architecture where many servers are configured in failover.

[2] describes the manifestations of webdefacement as well as their derived damages and proposes the prototype of a solution for the detection of webdefacement developed in C#. Their approach first consists of assigning priority to every page of a website based on their importance the home page being assigned the highest priority. Secondly, the system computes the hash code of every page which it stores in a database. Based on the priority assigned, each page will be checked with a frequency corresponding to its priority and the hash of that page at the moment of the check is compared to the one stored in the database if there is a mismatch it means that the page has been changed. They also proposed a Diff algorithm that computes the difference between two states of a page in order to spot the defacement. Finally they conducted a comparative analysis of their approach against other approaches and they came to the conclusion that their approach is more efficient than the others though it was only relevant for static pages.

[3] first analyses existing approaches that deal with webdefacement and from that analysis they noticed that none of them cope efficiently with the dynamism of actual web pages. They then proposed a solution inspired from anomaly detection in IDS to detect webdefacement by proceeding in two

main steps learning phase and monitoring phase. The learning phase consists of building a profile of a website in order to characterize the normal state of a website while the monitoring phase consists of taking a snapshot of the site at a predefined frequency and analyze it with sensors. Each sensor deals with a specific element of a web page including tags, word frequency, images, etc. The results of the analysis of the sensors are then transmitted to aggregator which in turn compute them in order to assess the difference between them and the page profile gathered during the learning phase and determine whether a webdefacement has occurred or not based on threshold values. They finally analyze reliability of their system especially false positives and false negative rates based on experiments.

Though [2] and [3] present interesting approaches to detect webdefacements, they have some flaws namely they do not address efficiently the dynamism of modern webpages and thereby generate many false positives and false negatives and also they cannot learn from new types of webdefacement attacks and improve their detection process over time.

Therefore, this paper is intended to provide an intelligent approach to efficiently detect webdefacement, identify the signature of webdefacement attacks and self-improve the detection based on new types of webdefacement and even legitimate updates of websites.

### 3 Research problem

Given the surge in webdefacements and the ever-growing importance of websites for companies, individuals and governments, tools that can detect webdefacements in an efficient and scalable way are highly needed. Some approaches have been developed to address this issue namely the one that consists of taking a snapshot of a page at regular intervals and comparing them with a baseline. However though this approach detects webdefacements it generates too many false positives as any change on the website including legitimate ones will be detected as webdefacements. This will generate too many fake alerts and thereby jeopardize the scalability and efficiency of the system. In order to handle this issue, we propose an approach based on artificial intelligence that will permit the development of intelligent algorithms that will learn the normal behavior of websites and efficiently detect webdefacements while minimizing false positives.

### 4 Webdefacement

Webdefacement is a cyberattack that can be defined as the unauthorized alteration of the content of a website. As a matter of fact and according to a report published by ITU-IMPACT on October 2014, webdefacements accounted for 22.76% of cyberattacks committed around the world which place it at the third position of cyberattacks ranking.

Traditionally to deface a website, attackers use some vulnerabilities or attack vectors like cross-site scripting (XSS), cross-site request forgery (XSRF), sql injection. But in recent years, hackers have started using DNS attacks to deface websites. By using this attack vector, they no longer need to attack the server that host the website, they simply gain access

to the DNS and tamper with its records in an effort to redirect users to another website under their control. The motivation that drives hackers to deface web sites used to be showing off their skills but in recent years a new driver came to the fore namely hacktivism which appears to be one of the most prominent motivation underlying webdefacements nowadays. Hacktivism can be defined as the act of hacking into a computer network or website for political purposes. Its main manifestations consist of changing the home page of government websites to diffuse propaganda. However, webdefacement can also be used to cover a more sophisticated attack or propagate malware: in fact hackers can use a drive-by download attack that consist of changing the content of a website so as to insert malware that will install automatically and stealthy into computers of every visitor who accesses the said page.

The figure below presents the defacement by an “Anonymous group” of the website of one of the leading Singapore newspaper:



Figure 1: Webdefacement illustration

### 5 Anomaly detection

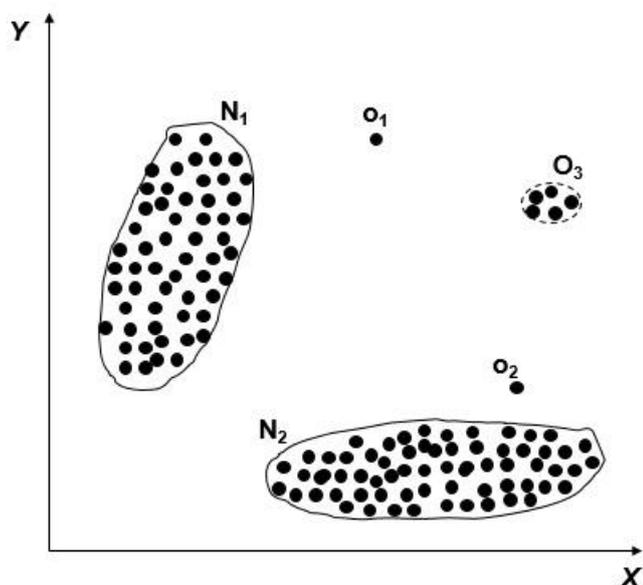
In recent years, with the digitization of almost every domain (Healthcare, finance, education, transport, etc) and the development of Internet of things, a huge volume of data are being generated by devices such as sensors and the need to detect anomaly from this data is becoming more crucial. An anomaly can be defined as a pattern in the data that does not conform to the expected behavior. Anomaly detection is very useful in a wide range of domains including healthcare, fraud detection, industrial damage detection, image processing and intrusion detection.

Figure 2 below provides an illustration of anomalies.  $N_1$  and  $N_2$  are region of normal behavior while  $O_1$ ,  $O_2$  and  $O_3$  are anomalies. Although anomaly detection may seem simple as just defining a region of normal behavior and tag as anomaly all data that doesn't belong to that region, it faces some challenges namely:

- the definition of “normal behavior” is not static as behavior trends may change over time ;

- the border between normal behavior and anomalous behavior is not always well defined depending on the domain ;

- Malicious adversaries may find ways to make anomalies look like normal behavior.



**Figure 2 : Anomaly illustration**

Anomaly detection techniques can output two types of data:

- label: a label (anomalous or normal) is assigned to an event or test instance ;
- score: a score related to the extent to which the test instance is considered anomalous is assigned.

Anomalies can be classified into three types:

- Point anomalies: They refer to individual data instance that are different from the rest of data ;
- Contextual anomalies: They refer to data instances that are anomalous within a specific context. Thus a contextual anomaly can be an anomaly in specific context and a normal behavior in another context. In this type of anomaly, data instances are usually defined using two set of attributes: contextual attributes that are used to specify the context of that data instance and behavioral attributes that are used to specify the non-contextual attributes of that instance ;
- Collectives anomalies: They refer to a collection of related data instances that is considered anomalous with respect to the entire data set while individual data instance composing this collection may be normal behavior.

Over the years, many anomaly detection techniques have been developed namely [6]:

- Classification based techniques: It consists of building a model that can classify normal behavior and anomalous behavior from learning through labeled training data. While it is very accurate in detecting known anomalies it requires high accurate data labels for various types of normal behavior which is often difficult to obtain. This techniques relies on concepts like neural networks, Bayesian networks and support vector machines ;
- Nearest neighbor based anomaly detection techniques: This technique decides whether a data instance is an anomaly or not based on the distance between that data instance and others. It can use the distance between the data instance and its  $k^{th}$  nearest neighbor or the distance between the data instance to all the other data instances. Though this technique is efficient and flexible as it can adapt to any type of data, its computational complexity can be problematic for huge and complex data sets ; Also it can generate false positives if the normal data instance doesn't have enough close neighbors or anomalous data instances have too many close neighbors;
- Clustering based technique: This technique first consists of grouping data into clusters. A data instance can be identified as an anomaly based either on the fact that it belongs to a large or dense cluster, or to the distance that separates it from the centroid of the cluster. While this technique offers some advantages as it can operate in an unsupervised mode and can adapt to different data types, the computational complexity of clustering data can be very high and the clustering algorithm can fail to capture the cluster structure of normal data instances.
- Statistical anomaly detection techniques: This technique relies on the construction of a mathematical model of the data set usually stochastic. A data instance is then identified as an anomaly if it occurs in the low probability region of the model previously built. The identification of anomaly is usually carried out through a hypothesis test statistic method. Though this techniques offers some advantages like the fact that it can operate in a unsupervised manner and provides an evaluation of the extent to which a data instance may be considered normal or anomalous it also poses some issues as it relies mainly on the assumption that data fit a particular statistic distribution which is not always the case and even if it was, choosing the best hypothesis test statistics might be quite difficult.
- Information theoretic anomaly detection techniques: this techniques relies on the analysis of information content of a set of data using information theoretic measures such as Kolomogorov complexity and entropy. A data

instance is then identified as an anomaly if it significantly alters the information content of the data set. Although this technique can be quite efficient as it can operate in an unsupervised manner and doesn't make any assumption about the underlying statistical distribution of data it also raises some issues namely their performance depends on the choice of information theoretic measure and it may be hard to provide an evaluation (score) of the extent to which a data instance may be classified as a normal behavior or an anomaly.

## 6 Our Solution

### 6.1. Methodology

In an effort to provide an efficient solution for the detection of webdefacement that will be less prone to false positives and false negatives, we designed a new architecture based on the following methodology:

1. Identify relevant criteria that can characterize the specificity of a web page and develop a module that will learn normal behavior of web pages based on those criteria ;
2. Develop a module that will learn defacement behavior through past example of web defacement so as to identify the signature of hackers ;
3. Develop a module that will crawl web pages and extract relevant data that will be used to identify web defacement ;
4. Develop a module that will assess data gathered from a web page in an effort to determine whether it has been defaced or not ;
5. Develop a module that will emit alerts by Email and SMS upon detection of web defacement ;

### 6.2. Description of the system modules

With regard to the methodology presented in the previous section, our framework is made up of five (05) main components namely Normal behavior trainer, webdefacement trainer, web crawler, Data processing and Alert which are depicted in the figures below. These modules will be described in subsequent sections.

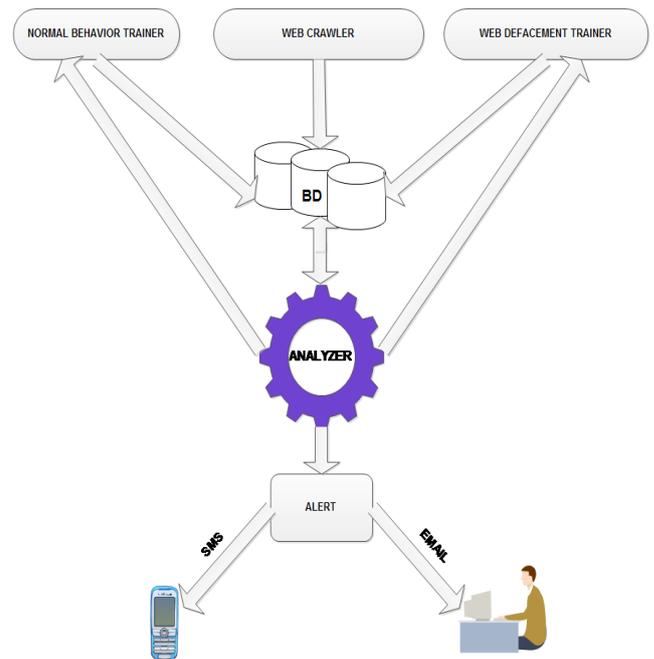


Figure 3: Architecture of our system

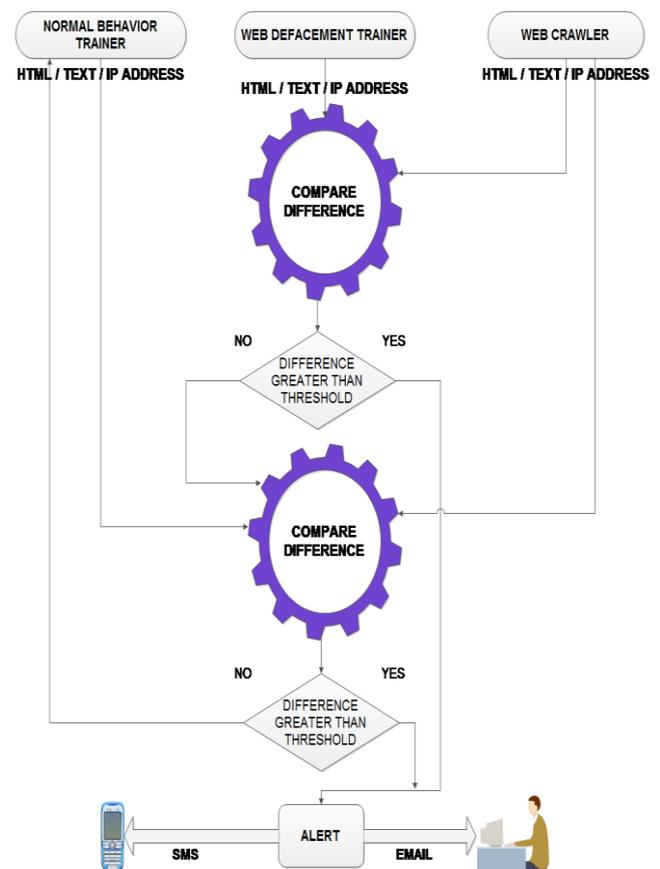


Figure 4: html/text/IP address processing by analyzer



## 6.2.1 Normal behavior trainer

This module is in charge of training the application to recognize the normal behavior of a particular web page.

In this light, the training process is based on some attributes that characterize the specificity of a web page. In our approach we identified the following attributes:

- The html structure of the web page: Since a website usually has a uniform structure for all its pages the html structure or arborescence is a critical element that can characterize the specificity of a web page ;
- The graphic charter of the page: Since a web page or website has a predefined set of colors, fonts, fonts size, font color these elements can characterize that particular web page
- The vocabulary of the page: A web page usually deals with a particular topic, therefore a topic can characterize the specificity of a web page ;
- The IP address of the host machine: A website is hosted on a computer that has a specific IP address. Since by hacking a DNS server a hacker can alter the host machine IP and thereby cause a webdefacement, this attribute constitutes a pertinent characteristic of a website. Because usually a website is hosted by the same company or by companies located in the same area, the IP address owner's name (obtained by the WHOIS) as well as its geographic location will be analyzed.

## 6.2.2 Webdefacement trainer

This component is aimed at assessing concrete examples of previous webdefacement in an effort to identify the signature of an attacker and confirm a webdefacement.

In fact, usually every hacker or group of hackers has a particular signature which can be an image, an identifier or sentence. This module goes through a defaced page in order to spot the image or text that have been added by the hackers. These elements will be stored in a database and if they appear in another web page later, it can indicate that that webpage has been defaced by the same hacker.

## 6.2.3 Web crawler

This component is aimed at crawling through pages of a website with respect to a frequency defined by the administrator. When this module crawls a web page, it extracts some data that will be useful in determining whether that page has been defaced or not. The data include:

- The html structure of the web page ;
- The text of the page ;

- The IP address of the host machine.

## 6.2.4 Analyzer

This component assesses information provided by the webcrawler module and compares them with the data obtained from the normal behavior trainer and webdefacement trainer as follows:

- The html structure of a web page at a given instance is modeled as a graph (tree); the set of all the structure of the web page at different moment are treated as a time series of graph. We then used an anomaly detection technique in dynamic graph called *feature-based* [1] to detect whether a particular structure is an anomaly or not and to spot the anomaly in that structure. A score that expresses the extent to which the anomalous structure differs from the normal trend is evaluated. If the score exceeds a defined threshold value, it indicates that a webdefacement has occurred. However, if the score is not null and doesn't exceed the threshold value, the html structure of the webpage at that specific moment is processed by the *normal behavior trainer* module in an effort to improve the normal behavior model of the webpage so that if a similar structure appears in the future it will be classified as normal behavior ;

- The text data extracted from the webpage is parsed and later analyzed using techniques presented in [5] that based on semantic network and tools like WordNet, permits to determine the key concepts and domains a block of text relates to. The key concepts extracted from a webpage at a particular moment are then compared to those obtained during the training phase and a score indicating the extent to which these key concepts are different is output. If the score exceeds a defined threshold value, it indicates that a webdefacement has occurred. If the score is not null and doesn't exceed the defined threshold value, the key concepts of the webpage at that specific moment is processed by the *normal behavior trainer* module in an effort to improve the normal behavior model of the webpage so that if similar data appears in the future it will be classified as normal behavior ;

- The IP address of the server that host the website at a particular moment is compared to those obtained during the training phase and a score that expresses the degree of difference between them is output. To evaluate the difference, this module takes into consideration several parameters namely the owner of the IP address obtained using the WHOIS and the geographic location of these IP. This evaluation is done using the anomaly detection method named *Nearest Neighbor distance*. If the distance output by the *Nearest Neighbor distance* method exceeds a defined threshold value, it indicates that a webdefacement has occurred. If the distance is not null and doesn't exceed a defined threshold value, the key concepts of the webpage at that specific moment is processed by the *normal behavior trainer* module in an effort to improve the

normal behavior model of the webpage so that if similar data appears in the future it will be classified as normal behavior.

It is worth mentioning that, whenever the system reports a webdefacement or normal behavior and the administrator notices that it wasn't the case, he notifies the application through the Graphic User Interface (GUI) so that the normal behavior trainer module learns from it in case it was definitely a normal behavior or the webdefacement trainer module in case it was definitely a webdefacement. This enables our system to constantly evolve and improve over time.

## 6.2.5 Alert

This component is aimed at delivering the result of our system to webmasters through different channels. When a webdefacement is detected, this module designs an alert message and sends it through SMS and Email to the website administrator.

## 7 Conclusion and future work

Due to the widespread use of the Internet and the development of E-business, websites have become very important for governments, companies as well as for individuals. These websites are used for many purposes namely E-commerce, blog, social network, E-government, marketing. However, because of their ever growing importance, websites have become one of the primary targets of hackers. Among attacks perpetrated by hackers against websites, webdefacement happens to be one of the most popular. Webdefacement which consists of altering the content of a website in an unauthorized way usually relies on several threats including XSS, XSRF, sql injection. The detection of web defacement has been a great concern over the past couple of years for engineers and scientist which led to the publication of some articles related to webdefacement detection approaches. However these approaches faced some challenges namely the regular occurrence of false positives due to the dynamic nature of modern websites and the fact that some websites can deal with several complex topics. This complicates efforts to differentiate between a legitimate change on the website and an illegitimate one. In order to overcome this issue, we propose in this paper a new approach based on artificial intelligence and anomaly detection. Our approach consists first of capturing the normal behavior of a webpage, crawl webpage at regular intervals, extract relevant information from the web page such as its html structure, the semantic of the web page then compares these elements to those related to the normal behavior of the webpage in question in order to determine whether there is a webdefacement or not and finally in case of webdefacement compare the signature of the attack to the previous ones (if any) and then stores the signature in case it appears for the first time. The innovation of this approach is twofold: first the continuous learning process as every time a web page is crawled the normal behavior model is improved

and this reduces false-positive and false-negative, secondly this approach takes into consideration three main criteria including the html structure, the semantic of the website content and the host server which considerably optimize the reliability and the efficiency of our approach. Future work can include the development of a prototype of a system that will implement this approach so as to evaluate its pragmatism, applicability as well as its computational complexity.

## References

- [1] Leman Akoglu, Hanghang Tong, Danai Koutra, "Graph based anomaly detection and description: A survey " in *Data Mining and Knowledge discovery*, 2014.
- [2] Tushar kanti, Vineet Richariya, Vivek Richariya "Implementation of an efficient web defacement technique and spotting exact defacement location using Diff algorithm " *International Journal of Emerging Technology and Advanced Engineering, Volume 2 Issue 3, march 2012*.
- [3] Eric Medvet, Alberto Bartoli, "Techniques for large-scale automatic detection of website defacement" PhD thesis, *Università degli Studi di Trieste*, 2008.
- [4] A. Cooks, M. S. Olivier, "Curtailling web defacement using a read-only strategy " in *Proceedings of the Fourth Annual Information Security South Africa Conference (ISSA2004)*, 2004.
- [5] Alessio Leoncini, Fabio Sangiacomo, Paolo Gastaldo and Rodolfo Zunino. "A Semantic-based framework for summarization and page segmentation in web mining", *Theory and applications for advanced text mining*, 2012.
- [6] Varun Chandola, Arindam Banerjee, Vipin Kumar. "Anomaly Detection: A survey" *ACM Computing Surveys*, 2009.

## Biography

**Dr. EBOT EBOT ENAW** obtained his B.Eng hons degree from Liverpool University in Electronic Engineering in 1989. He later obtained an M.Eng degree in Telecommunication Engineering from The University of Manchester England in 1991. He returned home where he was recruited in the University of Yaounde I, as an assistant lecturer. He pursued his university studies and obtained a PhD in Computer Sciences from the National Advanced School of Engineering of the University of Yaounde I, where he is currently a senior lecturer. His area of specialization include: computer network security, cryptography and formal specification and verification; theorem proving and model checking.



He has published many research articles in peer-reviewed international journals. In 2006 he was appointed Director General of the National Agency for Information and Communication Technologies Cameroon, a position he occupies till date. Major activities of the agency include amongst others: securing the Cameroon cyberspace through three key services: Computer Incidents Response Team (CIRT), Public Key Infrastructure (PKI) and Computer Security Audits.

**DJOURSOUBO PAGOU Prosper** obtained his Master degree in Computer science engineering from the National Advanced School of Engineering of the University of Yaounde I in 2009. He holds several certifications in networking and cybersecurity namely CCNA, CCNP, CEH, ECSA. In 2013, he was appointed subdirector of the National Computer Incidents Response Team (CIRT) of Cameroon, a position that he occupies till date.