Web Defacement Monitoring Tool
Project Description Document

J.Hou
3565155
Robert Sobukwe Rd, Bellville
Cape Town, 7535
3565155@myuwc.ac.za

ABSTRACT

This report focuses on development of a tool to minimize damage due to website defacements. The report follows the format of historical defacement events and other studies in the same field. The user requirements analysis states the web administrator perspective and the requirements analysis is broken down into system comments and software elements.

This project is sponsored by CSIR to educate users about web defacements, the effect and mainly develop a tool to prevent such hacktivism from happening.

KEYWORDS

Web defacement, cyber security, data breach, denial of service, WDMT.

1 INTRODUCTION

The advancement of the Web and the applications inspired new methods to communicate and share information, nearly all businesses and organizations utilize web pages to communicate with end-users. The Internet contains vast amount of web pages and information, a web page is a Hypertext Markup Language (HTML) document, typically a web site has many web pages linked, hosted on a web server [1].

A website contains information and critical data related to the organization, this typically gains audience from end-users and unwanted attention from hackers. The earliest examples of hacktivism and web defacement date back to 1996, the United State Department of Justice web server was hacked and defaced, hackers replaced the US Department of Justice website homepage with a text "Department of Injustice" and display of pornographic content [2]. More recent examples are, 2018 July 7 a report by News24 the Presidency government website was hacked and defaced by a hacker Black Team, the website was change with the text "Hacked By Black Team. Sahara is Moroccan. And Morocco is ur Lord!" [3] [4].

Web security are crucial to protect organizations in this digital era, hackers are improving exploiting any possible loop hole, its essential to practice and use web defacement monitoring tools.

2 LITERATURE REVIEW

Ebot Ebot Enaw and Djoursoubo Pagou Prosper proposed a Conceptual Approach to Detect Web Defacement through Artificial Intelligence [5]. The authors discuss the use of artificial intelligence concepts such as anomaly detection, machine learning and inferences to detect web defacement and unauthorized access [5]. The authors provided an intelligent way to efficiently detect the signature (type) of a web defacement attack, the detection algorithm is consistently self-improved [5]. The authors designed a new architecture to learn criteria that is used to characterize websites, through this the tool studies the behaviors of a normal web site [5]. A web defacement trainer module was developed to analyze previous web defacement signatures, and based on these signatures improves the accuracy of a web defacement attack [5]. The web crawler and analyzer module work in cohesion, the result is then compared to the result of a normal behavior module through given criteria [5]. The detection algorithm sends an email notification to the web administrator with the log if a web defacement is confirmed [5].

Tushar Kanti proposed Implementing a Web Browser with Web Defacement Dection Techniques [6]. The author discussed through the use of detection techniques and developed a web browser to enhance the detection accuracy of web defacement attacks [6]. The checksum is calculated through comparison of current webpage with the backed-up website, if there's a difference in checksum, this means defacement occurred. The difference algorithm is called when defacement is detected, the functionality of difference algorithm is to spot the exact location of the defacement, comparing with the original backed up version [6]. The web browser was modified based on Internet Explorer, the user will see no difference while use, the web browser will notify the web administrator of any detection of web defacements [6]. The author concludes the report with a recovery mechanism for the defaced web pages, the use of a difference algorithm to spot the exact location of defacement and the use of a checksum to replace the defaced html code [6].

3 PROJECT PROPOSAL

Web defacement is defined as unauthorized changes to a website such as text, company logos, images, and videos. In some cases, the entire website is completely changed. The perpetration is usually wanting to distribute a message (advertisement, malware link, etc.) or they may simply make fun of the website owner. In some cases, the defaced website may display offensive messages or information to viewers and customers. Web defacement is considered a trivial crime however it has the following implication on an organization and sometimes with lasting effects: humiliation and damage to reputation, services disruption and information downtime and potential data breach.

The objective of the project is to prevent web defacement through constructing a Web Defacement Monitor Tool (WDMT). The web administrator should install the corresponding tools, the WDMT will scan web file contents and its media, compare to the backed-up files by the use of hashing and checksum calculation. Detected defacement will be logged and send to the web administrator via email in report form, the WDMT will restore the damaged content from the backup, a further report will be sent after the restoration, logging defacement details.

4 USER REQUIREMENTS DOCUMENT

4.1 About the Project

The internet consists of vast amount of information, a common way to access this information is through the website. These days many users make use of websites, popular websites attract unwanted attentions for hacktivism.

This project was proposed by Council for Scientific and Industrial Research (CSIR), collaborative work with University of Western Cape (UWC) honor students. The aim of this project is to educate the user about what is web defacement, develop a web defacement monitoring tool and recover defaced sites by backups, the tool developed should be automated, identified defacement should be notified via log or screenshot in a document report form.

4.2 User view of the Project

The web administrator regulates and maintains web systems. In this project the user, the web administrator, must protect the website against defacement, defacement have negative impact on an organization, lost in public reputation, system down time and potential data breaches.

It is essential to have counter measurements against defacement; the objective of this project is to create a defacement tool that automatically detects web defacement and repairs damage content in a reasonable time.

4.3 Problem Domain

The problem domain in this project is understanding the HTML and web hosting, the website might be host online through a service provider, the user should be familiar with the terms and operations needed to implement the web defacement monitoring tool.

4.4 Expectations

The WDMT is expected to minimize damage due to web defacement on an organization this is accomplished through making a backed-up version of the web site. The WDMT is an automated tool that checks the current web site with the backed-up web site, ensuring security and quality of the web site. The WDMT should automatically repair damaged content once web defacement was detected, appropriate reports will be sent to the web administrator during detection and after repair phase of the tool.

4.5 Limitations and Out Scope

The WDMT doesn't perform the following functionalities:

- Track access information of end-user, such as IP address, MAC address and other system information.
- Reverse tracking and attempt to access end-user privacy.

4.6 User requirements

The user must do the following to ensure the software is operated correctly:

- Web Admin must create website with media content such as video and images.
- Install and implement WDMT.
- Using the WDMT, backup website and content with hashing, comparing checksum method.
- Set WDMT defacement automated scan interval example: 10 minutes.
- Set profile information such as name and email.

4.7 User Case Diagram

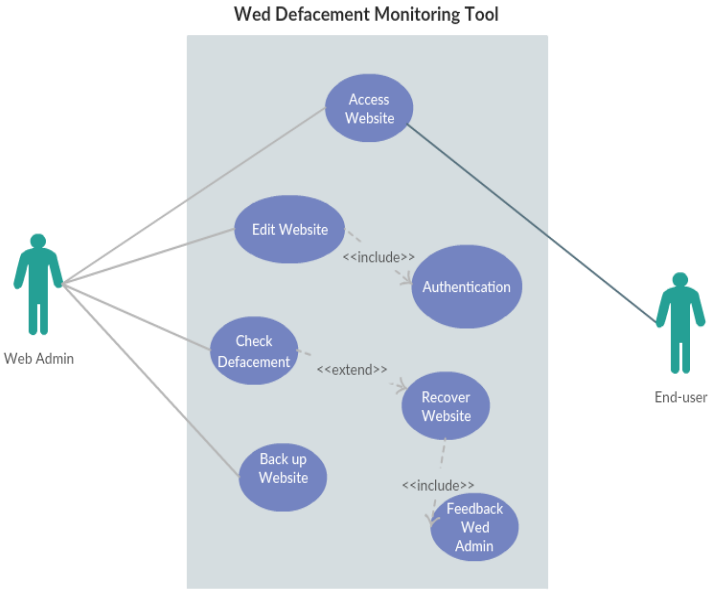

Figure 1 Use Case Diagram of WDMT

5 REQUIREMENTS ANALYSIS DOCUMENT

5.1 Current System

The WDMT is being developed on a Linux platform, prerequisite software's are Chrome or any web browser software, Python 3.5.2, GitHub and Git 2.7.4.

The figure below shows the Linux environment on the right and temporary website used to test against web defacement on the left.

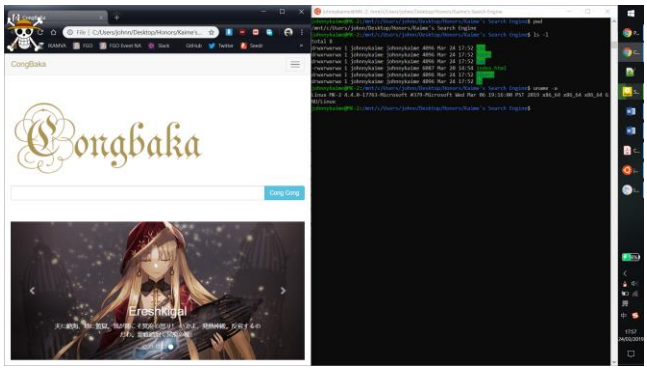

Figure 2 Current System used to develop WDMT

5.2 Stakeholder Requirements

In this project, the stakeholder CSIR defined the following requirements, the user group consists of web administrators, technician that create and maintain websites, the WDMT must include the following functionalities:

- Create backups of a website files contents, images, videos and web text
- Instantly detect when a website has been defaced and notify the web administrator
- Restore damage content in the shortest possible time.
- The mentioned functions of web defacement detecting tool should all be automated.

Other user groups are end users that test the WDMT or visit the web site for various information's, these users have no access to files for security measurements.

5.3 Functional Requirements

- The WMDT must back-up the current undefaced web site.
- The WDMT must detect web defacement attacks. Attacks include any modification or missing of text, graphic images or video.
- The WDMT must provide feedback in report form and email to the web administrators. First report when web defacement was detected.
- The WDMT must recover damage site through comparing hashed checksums with the backed-up website.
- The WDMT must provide a secondary feedback report after restoration of damage web sites.

5.4 UML Class Diagram

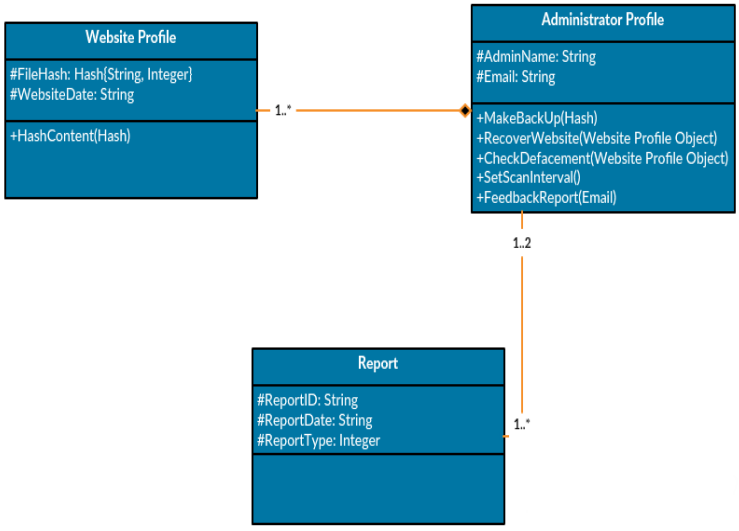

Figure 3 UML Class Diagram

6 NON-FUNCTIONAL REQUIREMENTS

6.1 Performance Requirements

The WDMT scan should be automated and each scan in attempt to identify defacement should not take longer than 5 minutes.

6.2 Operating Requirements

Compulsory software is Python 3 and Linux platform is required to operate the WDMT.

6.3 Reliability

Reliability to WDMT is a big concern, the WDMT should be always check input and filename to prevent human error, WDMT will prompt feedback if files of corresponding filenames are not found.

6.4 Usability

The WDMT will be a terminal line base software, the web administrator can interact the software using number such as 1, 2, 3 etc. Each number corresponds to a different functionality, a menu mapping will be provided in the command menu interface for more ease of access.

6 USER INTERFACE

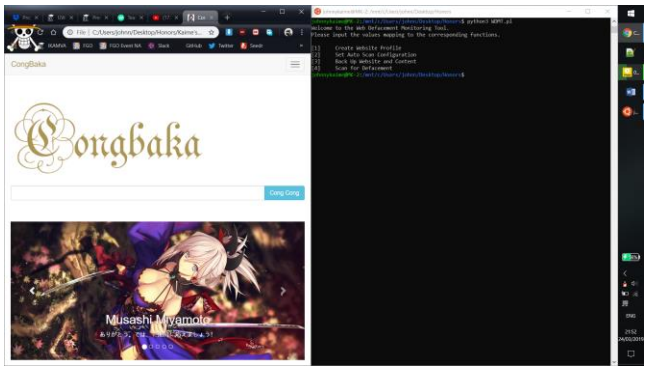

Figure 4 Interface of Website and WDMT

The web administrator is expected to create a website containing media such as videos and images. The above figure 4 shows a website used as a test against the WDMT.

The main interface for the WDMT is command line based, it operates through a terminal in Linux or Bash terminal on Windows 10. The web administrator will navigate the WDMT via numbers, in the above figure 4, the web administrator would have to enter 4 in the terminal to force scan any web defacements, provided all the necessary steps are completed.

REFERENCES

[1] M. Masango, F. Mouton, P. Antony and B. Mangoale, "Web Defacement and Intrusion Monitoring Tool: WDIMT," September 2017.

[2] D. Dorothy, "Georgetown Journal of International Affairs," *The Rise of Hacktivism,* 2015 September 2015.

[3] M. Mxolisi, "Presidency website up and running after hacking attack," News24, 7 July 2018. [Online]. Available: https://www.news24.com/SouthAfrica/News/breaking-presidency-website-hacked-20180707. [Accessed 6 March 2019].

[4] N. Mphathi, "SA Presidency website hacked," Independent Online (IOL), 11 July 2018. [Online]. Available: https://www.iol.co.za/dailynews/news/sa-presidency-website-hacked-15950026. [Accessed 6 March 2019].

[5] E. E. Enaw and D. Pagou Prosper, "A Conceptual Approach to Detect Webdefacement Through Artificial Intelligence," *International Journal of Advanced Computer Technology (IJACT),* vol. 3, no. 6, pp. 77-83.

[6] K. Tushar, "Implementing a Web Browser with Web Defacement," *World of Computer Science and Information Technology Journal (WCSIT),* vol. 1, no. 7, pp. 307-310, 2011.

TABLE OF FIGURES

APPENDIX

A Project Plan Gantt Chart