Analyzing the Security Posture of South African Websites

J. Mtsweni
Defence, Peace, Security and Safety (DPSS)
Council of Scientific and Industrial Research (CSIR)
Pretoria, South Africa
jmtswei@csir.co.za

Abstract— **Today, public-facing websites are virtually used across all different sectors by different types of organizations for information sharing and conducting core business activities. At the same time, the increasing use of mobile devices in Africa has also propelled the deployment and adoption of web-based applications. However, as the use of websites increases, so are the cyber-attacks. Web-based attacks are prevalent across the globe, and in South Africa an increase in such attacks is being observed. Research studies also suggest that over 80% of the active websites are vulnerable to a myriad of attacks. This paper reports on a study conducted to passively analyze and determine the security posture of over 70 South African websites from different sectors. The security posture of the local websites was thereafter compared against the top ten (10) global websites. The list of the websites was mainly chosen using the Amazon's Alexa service. The focus of the study was mainly on the security defense mechanisms employed by the chosen websites. This approach was chosen because the client-side security policies, which may give an indication of the security posture of a website, can be analyzed without actively scanning multiple websites. Consequently, relevant web-based vulnerabilities and security countermeasures were selected for the analysis. The results of the study suggest that most of the 70 South African websites analyzed are vulnerable to cross-site scripting, injection vulnerabilities, clickjacking and man-in-middle attacks. Over 67% of the analyzed websites unnecessarily expose server information, approximately 50% of the websites do not protect session cookies, about 30% of the websites use secure communications, in particular for transmitting users' sensitive information, and some websites use deprecated security policies. From the study, it was also determined that South African websites lag behind in adopting basic security defense mechanisms when compared against top global websites.**

Keywords-cybersecurity; websites; websecurity; web applications; security policies, south africa;

I. INTRODUCTION

In 2007, Acunetix [1] revealed that on average 70% of the public facing websites are vulnerable to different types of attacks, such as SQL injection. Other studies suggest that over 80% of government websites around the world are vulnerable to common web-based attacks [2]. In 2012, three (3) South African government websites were hacked [3], although the government expressed lack of concern, since they claimed web servers hacked had no sensitive information. A year later, WhiteHat Security [4] reported that at least 86% of global websites that they have studied had one or more serious vulnerability, which could lead to different types of web-based attacks. The study further reported that at least one (1) website had over 50 vulnerabilities with cross-site scripting being the most common and serious one [4]. In 2011, another report revealed that on average, a website has about 79 serious vulnerabilities [5].

Recently, official news reported that over 50 South African websites were simultaneously hacked [6], including a website of a large corporation (i.e. Sasol) [7]. Some of the South African websites (e.g. ANC) have been hacked more than once over the past few years, with a recent attack recorded early in 2015 [8]. Using the zone-h archive service (i.e. www.zone-h.org), it was also determined that within a week, over 171 .co.za websites were defaced, and close to 90,000 .co.za websites have been defaced since 2002.

To emphasize that websites' security is a global challenge, in France, hackers targeted, using denial of service (DoS) attacks, about 19,000 websites in January 2015, which experts suggest was a response to demonstrations against Paris terror attacks [9]. During the recent 2015 elections in Nigeria, a website of the electoral commission was also hacked. It is therefore clear that some of these websites are exploited by attackers for various reasons including data theft and political agendas.

The aforementioned reports seem to suggest that websites and related web applications are considered the weakest link by adversaries. As a result, a number of server-side and client-side security policies are continuously being devised and improved to minimize some of these web-based attacks. The unfortunate part is that most websites developers are slow or ignore to implement available and appropriate security countermeasures on their websites [10]. Thus, many websites still fall victim to known vulnerabilities, purely because secure web development practices are not followed.

The main purpose of this paper is thus to report on a study conducted to passively analyze and determine the security posture of over 70 South African websites from different sectors. The study was conducted with an objective to raise awareness regarding the countermeasures that could be deployed to strengthen the security posture of public facing

978-1-4799-7755-0/15/$31.00 ©2015 IEEE

websites.

The remainder of this paper is structured as follows: Section II discusses the research approach followed to conduct the study, and explains how the 70 websites were selected. The common web-based vulnerabilities and attacks considered for the analysis are discussed in Section III. Section IV highlights the basic client-side security policies that could be employed to minimize some of these common vulnerabilities. In Section V, the results of the study are presented highlighting the security posture of South African websites and comparing the results against the security policies implemented by the top global websites. In Section VI, related work is briefly highlighted with the aim to demonstrate the significance of the work presented in this paper. In section VII, the paper is concluded by discussing some possible remedies that could be implemented to improve the security posture of South African websites.

II. RESEARCH APPROACH AND DATA COLLECTION

In this section, we discuss the research approach adopted for this study and the techniques used for collecting data from the selected websites.

A. Research Approach

There are various methods used today to test the security of web applications [10]. Some include: vulnerability and penetration testing, mutation testing, graph-based testing, and others. These techniques perform security testing by actively scanning and injecting specially crafted inputs into the web pages and observe the resulting behavior, which provides an indication if a website is vulnerable or not to specific vulnerabilities. Such an approach is very useful, because multiple websites can be tested for multiple vulnerabilities within a short space of time. Nevertheless, the approach of actively scanning websites is often viewed by website owners as an abuse of their systems, unethical and illegal; unless permission to actively scan had been obtained [2].

This study adopted a passive scanning approach, which focuses on analyzing and auditing the security policies that are presented to the browser via HTTP response headers by the web server during invocation. According to [11], these security policies make it simpler to determine websites' security through passive analysis. It is further noted that although the client-side security policies are not a solution to all web security vulnerabilities, their adoption can indicate a positive security posture of a website [11].

B. Website selection and data collection

In order to choose the websites for the analysis, we consulted the Amazon's Alexa service [12], which provides analytical insights regarding global and local top websites. We also corroborated the information retrieved from Alexa with the study published by Effective Measure on the top 20 South African websites [13]. We further consulted the South African National Government portal (i.e. www.gov.za), which includes a list of all government websites that tend not to appear in the top websites ranked by commercial services. From the Alexa and Effective Measure lists of top websites, we then selected 30 South Africa websites, mainly commercial spreading across different sectors, such as news, sports, e-commerce, and banking. A list of 40 government websites from the government portal was additionally extracted leading to a total of 70 websites studied.

Furthermore, we selected a list of top 10 global websites using only the Alexa service. This list was used to derive some of the security defense mechanisms for analysis, and to also compare against the security posture of the 70 selected websites. In addition, web-based security policies and vulnerabilities that could be passively analyzed were derived from related studies [5], [11], [14]. The vulnerabilities and security policies considered for the study are discussed in Section III and Section IV respectively.

The data used for the results presented in this paper were collected by manually loading the home URL address of each of the selected website on the browser (Google Chrome, Mozilla Firefox, and Internet Explorer). Thereafter, embedded browser developer tools were used to analyze the HTTP response headers, which include a number of security and non-security policies that need to be enforced by the web browser. All of the 70 websites were each analyzed three (3) times using three (3) different browsers, and this was important since the security posture of any website, including accessibility could change at any time, but also because different policies are supported differently by different browsers.

III. COMMON WEBSITE VULNERABILITIES AND ATTACKS

A number of technical reports and research studies [4], [15] are released on regular basis highlighting common website vulnerabilities. Although, it is suggested by [5] that a number of vulnerabilities per website had sharply decreased over the years, on average each website still suffers from approximately 79 serious vulnerabilities, which could ultimately lead to data loss, data corruption, and system interruptions.

The Open Web Application Security Project (OWASP) is one of the leading initiatives that strive to raise awareness about web security, particularly by identifying serious vulnerabilities facing by public-facing websites. Every three years, OWASP releases what they coined "OWASP Top 10" reports that highlight top serious risks in web applications, and at the same time provide basic techniques for affected organizations to protect themselves against the identified common vulnerabilities. The 2013 OWASP Top 10 report [15] highlighted ten (10) most critical web application security vulnerabilities that website owners need to protect themselves against.

Based on these reports, the following vulnerabilities were considered for the analysis of the selected websites.

- **Broken authentication and session management**: lead to passwords and sessions compromise due to poor implementation related to authentication and session management on web applications. For the study, we only focused on session management, which can be

strengthened by the set-cookie HTTP response header [16].

- **Cross-site scripting (XSS)**: occurs as a result of untrusted data being sent to a web browser without proper validation, leading to web-based attacks, such as user sessions hijacking and website defacements. According to [5], cross-site scripting are the most prevalent in websites.
- **Security misconfiguration**: this risk can occur at any level of a web application, and comes about due to lack of proper security hardening. For an example: unnecessary information disclosure (e.g. server software version), unnecessary software features enabled, use of deprecated features, and too much information in error messages.
- **Sensitive data exposure**: websites risks exposing sensitive information (e.g. authentication credentials) over the Internet via attacks, such as Man-In-Middle Attacks (MiTM) when using protocols (e.g. HTTP) that do not support encryption.
- **Using components with known vulnerabilities**: websites that use software or libraries with known security flaws expose themselves to a variety of attacks, and may further undermine existing security defense mechanisms.
- **Clickjacking**: is a technique used by attackers to trick oblivious users into clicking to malicious web pages, UI elements (e.g. buttons) or links. An attacker will execute this attack by overlaying a web page or UI elements (e.g. button) from a legitimate website with malicious code. This vulnerability occurs when a website is enabled, which is by default, to be externally loaded within iframes [17]. That is, Banking websites are generally the most targeted with this vulnerability.

The number one (1) web application vulnerability highlighted in the OWASP top 10 [15] is data injection, which occurs when a malicious actor craft special input or data in order to trick the web application into executing undesired actions (e.g. reading all database records). This vulnerability was not considered for our study, since it requires active scanning and analysis.

IV. CLIENT-SIDE SECURITY POLICIES

In appreciation of the continuous rise of web attacks, forward-thinking countermeasures are regularly being introduced, particularly for web browsers. One of the recent initiatives are the declarative security policies, which are defined by the website owners and enforced by the web browser as the web pages are invoked. According to [18], these client-side security mechanisms, which are presented via HTTP response headers, are purported to "compel browsers to perform specific security functions", and in turn protect websites and their users from different types of attacks. Admittedly, these security defense mechanisms are not foolproof for all the web security challenges, but play a very critical role in minimizing various web-attacks (e.g. XSS, CSRF, clickjacking, MIME sniffing and session hijacking), and overall improving the security posture of a website.

A number of these declarative security policies are emerging from different organizations, including Microsoft, Google, Mozilla, and mostly supported, recommended, and standardized by the Internet Engineering Task Force (IETF) and World Wide Web Consortium (W3C). Although not all web browsers support all of these security mechanisms, most are well established and adopted by top global websites, which indicate that they are worth considering for increasing the security of websites.

For the study presented in this paper, a couple of these security mechanisms were selected for the analysis. The selection was based on a number of factors, including adoption by top global websites (e.g. google.com and facebook.com), their relevance in countering the common vulnerabilities discussed in Section III, support across different browsers, and related work.

The following discussion highlights the security policies considered for this study

- **Secure communications (HTTPS):** the adoption of HTTPS by websites for securing transfer of information over the Internet, particularly to minimize sensitive data exposure, is growing. A number of commonly visited websites are implementing HTTPS and for the study, we analyzed the selected websites for HTTPS implementation, especially when sensitive information is involved, such as *Login*, *My Account*, *or Webmail* web pages are accessible on the website. It is fairly easy to check if a website implements HTTPS or not, however, for this study we did not focus on verifying if HTTPS implementation was solid, because it has been reported in various studies that some HTTPS implementations are flawed and vulnerable to attacks [11].
- **HTTP Strict-Transport-Security:** in order to avoid attacks, such as SSL stripping [19], website owners can implement the HSTS declarative policy [18] that would compel a web browser to only communicate with the server via an encrypted channel for a specified period of time. This security policy is important for websites (e.g. banking websites) that use both HTTP and HTTPS for communications. For the study, we specifically investigated if a website that uses HTTPS also implements HSTS as an extra layer of security.
- **Set-Cookie:** It has been demonstrated many times that adversaries can easily steal session cookies using various mechanisms, including exploiting unsecure channels and injecting malicious scripts into web pages. Thus, the Set-Cookie [16] security feature is very useful for session management and preventing XSS attacks in websites. It can be configured using two key values: *Secure*, which ensures that session cookies are only exchanged via secured channels and *HTTPOnly*, which compels the browser not to share cookies with any JavaScript code executing in the browser.

- **X-Frame-Options:** one of the most ignored and yet simplest web-based attack to prevent is clickjacking [20]. Thus, the X-Frame-Options [16] is meant to configure how a website, including its web pages and related elements are loaded into frames by third-party websites. The valid settings for this security header are: *DENY*- refuses any resource from framing any parts of the website; *SAMEORIGIN*- allows only resources that reside within the scope of the affected website to load web pages and related components into frames; and *ALLOW-FROM*- informs the browser to only frame the website from a specified trusted source. It is worth noting that this declarative policy will be replaced by the Content-Security-Policy, which is described in the following point.
- **Content-Security-Policy (CSP):** this directive policy is still being matured by Mozilla and Google, but its main objective is to consolidate and improve on some of the previously defined policies. The approach with this policy is to implement multiple client-side web-based defense mechanisms using one HTTP response header [14], such as combining X-Frame-Options and X-XSS-Protection.
- **X-Content-Type-Options:** this feature is meant to address Internet Explorer vulnerabilities that lead to MIME sniffing [18]. It is also used by other websites to protect against malicious attacks when downloading Google Chrome extensions. It only has one value, that is, *nosniff* [16], to prevent the browser from sniffing content types not specified by the web server. Thus, with this policy declared chances of drive-by download attacks [16] are minimized.
- **X-XSS-Protection:** this header was implemented by Microsoft to lessen the risks of reflective XSS attacks [21]. When set, this feature prevents the browser from explicitly rendering undesired content [18]. It is worth noting that this security feature is enabled by default in recent browsers, however, users are still able to disable it.
- **Server Information:** one of the oldest non-security HTTP response headers that are by default configured by many web servers is the *Server* response header [16]. Its purpose is to reveal the name of the server to the browser, including its version and it some cases modules that are installed on that specific server. The information exposed by this header is mostly useful for statistics purposes, such as those gathered by Netcraft. However, some of the information exposed by this declarative header is unnecessary and could provide a malicious user with vital information to attack. For example, an outdated and vulnerable web server can easily be exposed via this header. Thus, in the recent times, top global websites have started obfuscating the information sent via this specific response header in order to curb the attacks that arises as a result of misconfigurations and use of known vulnerable components. For the study, it was important for us to determine if the studied websites are unnecessarily exposing vital information to attackers using the Server response header.
- **Privacy statements:** generally, websites have long been using privacy statements and/or terms and conditions as a legally binding contract between the website owners and users. In today's world where web-based attacks that have the potential to compromise users' information are on the rise, privacy policies are even more important. In actual fact, privacy statements are mandatory by law in many countries for specifying how personally identifiable information (PII) is treated when collected via a website, whilst terms and conditions covers diverse range of user agreements, such as conditions of access, linking to 3rd party websites and so forth. For our study, we decided to also analyze the selected websites against the presence of such a security policy, because such information can also provide an indication of the security posture of a website.

Lastly, the client-side security policies discussed above have been implemented as mentioned by some top global websites. One relevant example of a website that widely uses most of these policies is the Facebook website as illustrated in Figure 1.

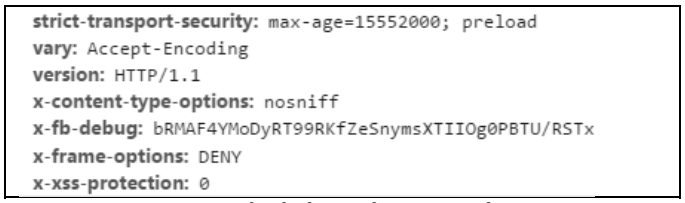

Figure 1: Facebook client-side security policies

V. SECURITY POSTURE OF SOUTH AFRICAN WEBSITES

In this section, we provide a detailed overview of the results from the assessment of the 70 websites. The results are presented based on the security policies highlighted in Section IV.

A. Website categories

As discussed in the research approach section, a total of 70 South African websites was selected for the study and the categories were spread as tabulated below.

TABLE I: WEBSITES DISTRIBUTION

Classified	3
E-commernce	2
Education	4
Banking	5
Government	40
Insurance	1
News	3
Research	2
Social	2
Telecoms	2
Sport	2
Jobs	2
Retail	2

The websites that were part of the *classified* category included websites that allow users to buy and sell their

products online. Because some of the studied websites had serious flaws during the study, we decided not to publish their names in this study for obvious reasons.

B. Secure communication

The results of the study indicate that only 23 of the 70 South African websites analyzed use HTTPS. It is worth noting that some of the websites that do not even use HTTPS to allow users to register, login and access web e-mails via their websites. This is problematic because username and passwords are sensitive information and should be protected using appropriate methods.

Furthermore, only 2 websites that used HTTPS had implemented HSTS to minimize SSL stripping attacks.

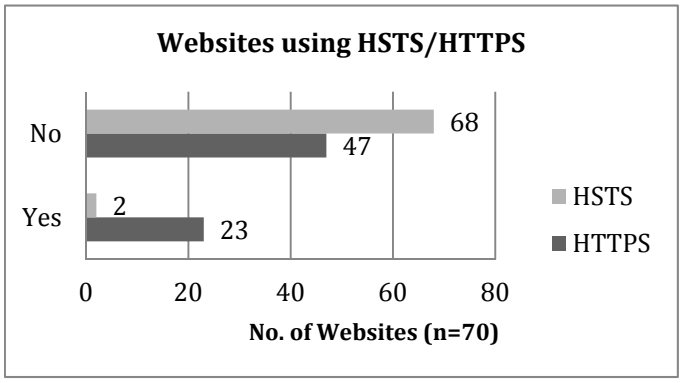

Figure 2: Websites that implement secure communications

Compared to the global websites, South African websites lag behind in using secure communications. Out of the 10 global websites used for the comparison, 90% of the sites implemented HTTPS and only 27% deployed HSTS.

HSTS mechanism, although implemented more than 2 years ago, its use by South African websites is almost non-existent. However, according [22], the problem is global, mainly due to the reasons that developers are not aware of its benefits and some browsers (e.g. IE) do not support it.

C. Secure session cookies

The results of the study show that most of the websites assessed have widely adopted the Set-Cookie HTTP response header. This should not be surprising since this header was introduced in the 90's by IETF [23], and has been improved over the years. As described in Section IV, an example of its implementation can be seen in Figure 3, which include the session cookie, path, domain, and two key values (secure and HTTPOnly), which are meant to secure cookies from malicious actors.

Figure 3: Twitter.com Set-Cookie policy

Our analysis of the 70 websites indicates that 74% of the websites use the set-cookie response header for session management. However, only 21% of the websites had the *HTTPOnly* setting in the set-cookie header, which protects the browser against sharing sensitive cookies with external JavaScript code. In addition, 53% of the websites employed the *Secure* option, which ensures that the session cookie is only transported via an encrypted channel.

When comparing our results with those of global websites, we found that 80% of the sites used the set-cookie response header. Furthermore, 73% of the global websites employed the *HTTPOnly* option and only 27% used the *Secure* option during the time of the study.

D. Clickjacking protection

As may be noted in Figure 4, 86% of the websites studied did not implement the X-Frame-Options. This basically means most are vulnerable to UI redressing attacks, which target oblivious users via trusted websites.

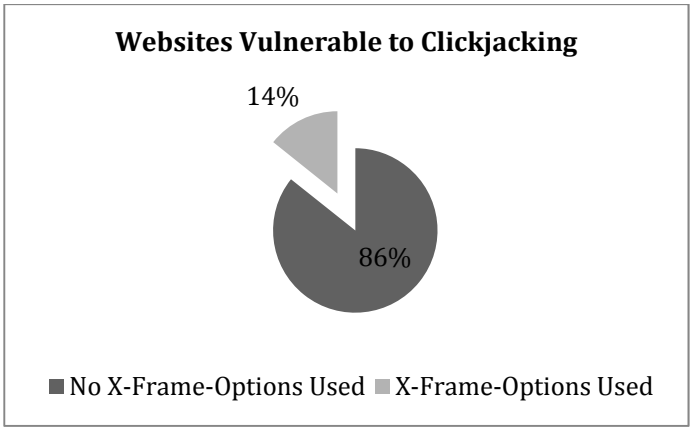

Figure 4: Websites vulnerable to clickjacking

Comparatively, 90% of the top ten global websites analyzed are protected from clickjacking through the implementation of the X-Frame-Options by either using DENY or SAMEORIGIN value.

E. Content-Security-Policy

The CSP policy as a W3C standard is emerging; it started being used by website owners in 2011/2012 and improvements are still being made. Based on the South African websites studied, only 3 of the 70 websites were found to be using this policy. These three websites are in government, retail, and banking sectors.

When comparing these results with the top 10 websites, it was clear that CSP is not widely adopted as only 2 global websites showed evidence of CSP implementation. The two websites are Facebook.com and Twitter.com. According to [14], as a defense-mechanism, CSP is rated higher than any other HTTP security response header; thus, it is a bit surprising that the header is not widely adopted.

F. X-XSS-Protection

None of the 70 South African websites analyzed were found to have explicitly implemented X-XSS-Protection. Nevertheless, it should be noted that XSS protection is enabled by default in most browsers, such as IE. Thus, this might be one reason why none of the South African websites directly employed it.

On the contrary, 55% of the global websites were found to have explicitly implemented the X-XSS-Protection HTTP header.

G. X-Content-Type-Options

This is another HTTP response header that is not widely adopted by South African websites. The low adoption could be attributed to the fact that it is mostly useful in IE. The results of the study revealed that only 1 website out of the 70 use this header to enhance browser security.

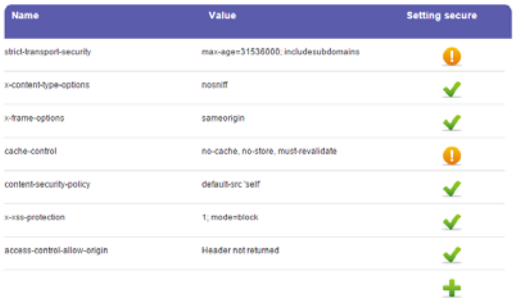

Figure 5: HTTP Response headers implementation

Comparatively, 36% of the global websites rely on this header to counter MIME sniffing attacks. Figure 5 above depicts an example of some of the data we retrieved for one global website, which suggest the adoption of almost all the security defense-mechanism highlighted in this paper.

H. Server Misconfiguration

The results of the study disclose that most of the websites predominantly use either Apache or IIS webservers, IIS having a slight edge (43%) and Apache (41%) (*cf.* Figure 6). At least 69% of the web servers (both IIS and Apache) had security misconfigurations leading to potential sensitive data exposure.

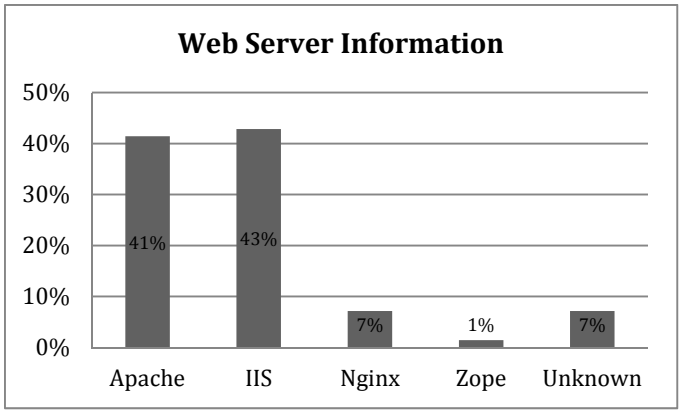

Figure 6: Webservers used by the websites studied

Interestingly, it was not straight-forward to determine the same statistics for the top 10 global websites, because many of these websites obfuscate the server information they send to user agents. We only discovered that 27% of the global websites use Apache, 9% are using Nginx, and 64% of the webservers information was not exposed. For this study, the non-exposure of server information is seen as a good security practice. This is even more relevant in minimizing web-based attacks. Figure 7 demonstrates unnecessary information sent via the Server HTTP response header.

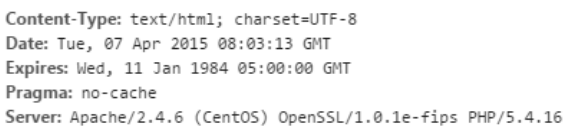

Figure 7: Server Misconfiguration

Websites that expose unnecessary server information tend to also expose server version numbers, which might indicate if the server software is vulnerable or outdated (e.g. Figure 8). Such information can be very useful to an attacker.

Figure 8: Outdated Apache Server Software

I. Privacy Statements

On the positive side, it was discovered from an analysis that about 74% of the websites studied incorporate privacy policies and/or terms and conditions on their home page. However, all (100%) of the top ten global websites had a link on their home page to a privacy policy and/or terms and conditions. What is worth noting is that in most cases the privacy statements were not consistent across the websites in the same sector. In one case, a privacy policy was plagiarized from a foreign websites and references to the foreign website were not changed. Most of the privacy statements dealt with the aspects related to security of personal information, interceptions, postings, prohibited use, children's privacy, disclaimers and other provisions.

Based on the results discussed in this section, it is apparent that South African websites can still improve when it comes to implementing client-side security policies that can be enforced by the browser. In addition, South African websites still lag behind on a number of aspects when it comes to securing websites. It should be highlighted that all the security mechanisms discussed in this study do not require heavy investment for implementation, but only proper server hardening before deploying the website. In most cases, the ISP (Internet Service Providers) could be the ones implementing most of the generic security headers, but at the same time software developers need to start considering security as a process of development and not an add-on activity.

As previously noted, most of the security mechanisms highlighted should not be seen as the first line of defense for all different types of web security problems, but should be viewed as means to strengthen web security and minimize the effects of cybersecurity attacks of websites.

The following section summarizes some of the studies that we considered related to our work. These studies were also consulted for deriving the security mechanisms used for the analysis.

VI. RELATED WORK

Based on our extensive search, no similar studies focusing on the South African environment could be found. However, similar studies have been conducted in the recent past focusing predominantly on the United States, Chinese and European websites.

In [14], an experiment was conducted following a crawling approach to analyze the security status of over 22,000 websites from 28 European countries. Some of the metrics used for the analysis are similar to those used for this study. This study [14] further investigated common vulnerabilities on the selected websites. The general results from this study suggest that at least 46% of the assessed websites had deployed one or more of the declarative security policies, which is not entirely the case in our analysis.

A study by [18] focused on surveying the adoption status of the client-side security policies in online banking websites. They investigated top 40-plus safest banks as evaluated by Global Finance, none of which were in South Africa. Only five (5) policies were used for actively analyzing the selected websites. The results of the study indicated that not even a single bank website, at the time of the study, had implemented any of the five (5) security policies.

Using a crawling experiment, a large-scale analysis of over 10,000 popular Chinese websites was done based on five (5) HTTP response headers by [11]. In addition, this study investigated the security of HTTPS implementations in the sampled websites. Generally, the results indicate that most Chinese websites lag behind in adopting appropriate client-side security policies and over 84% of the crawled websites had SSL/TLS implementation issues.

Lastly, a security assessment by [2] of over 50 U.S state e-government websites was conducted to "identify opportunities and threats for the sites and their users". The results of the study indicated that most of the websites implemented privacy statements and/or security policy statements and 98% of the studied websites used encrypted channel for transporting users' sensitive information.

VII. CONCLUSION

Web security is still an active research area, and organizations, such as W3C and IETF continue to develop standards that could contribute towards a better and secured web environment. The declarative HTTP headers are seen as possible solution towards inexpensively protecting the web and its users.

With over 90,000 South Africa websites (at least when using .co.za domain name) having been defaced since 2002, it is clear that many websites are not secured and as such have the potential to expose users' sensitive information. From the results of the study, it is also clear that most of the 70 South African websites studied are vulnerable to common attacks, such as clickjacking and cross-site scripting. This is mainly due to lack of implementation of existing solutions by web developers.

When comparing the results of the analysis with the global websites, it became clear that there is a room for improvement by South African websites, especially websites that are considered critical, such as banking.

This study was limited to 70 websites due to the manual approach chosen for the analysis. Thus, further research, using non-intrusive crawling experiments could be conducted so as to have a wider view of the security posture of South African websites.

REFERENCES

[1] Acunetix, "70% of websites at immediate risk of being hacked!," 2007. [Online]. Available: http://www.acunetix.com/blog/news/70-percent-websites-immediate-risk-hacked/. [Accessed: 08-Apr-2015].

[2] J. J. Zhao and S. Y. Zhao, "Opportunities and threats: A security assessment of state e-government websites," Gov. Inf. Q., vol. 27, no. 1, pp. 49–56, Jan. 2010.

[3] Mail & Guardian, "Three SA government websites hacked," 2012. [Online]. Available: http://mg.co.za/article/2012-12-09-three-government-websites-hacked. [Accessed: 08-Apr-2015].

[4] SC Magazine, "2013 Website security statistics report," SC Magazine, 2013.

[5] J. Grossman, "The State of Website Security," IEEE Secur. Priv., vol. 10, no. 4, pp. 91–93, Jul. 2012.

[6] Balancing Act, "Hacker targets South Africa Linux Sites," 2014. [Online]. Available: http://www.balancingact-africa.com/news/en/issue-no-156/web-and-mobile-data/hacker-targets-south/en. [Accessed: 08-Apr-2015].

[7] ENCA, "SA websites hacked," 2014. [Online]. Available: http://www.enca.com/technology/sasol-website-one-dozen-sa-websites-hacked. [Accessed: 08-Apr-2015].

[8] J. Vermeulen, "ANC website hacked?," myBroadband.co.za, 2015. [Online]. Available: http://mybroadband.co.za/news/security/119346-anc-website-hacked-2.html. [Accessed: 08-Apr-2015].

[9] SAPA-AFP, "Hackers target French websites," myBroadband.co.za, 2015. [Online]. Available: http://mybroadband.co.za/news/quick-news/116964-hackers-target-french-websites.html. [Accessed: 09-Apr-2015].

[10] Y.-F. Li, P. K. Das, and D. L. Dowe, "Two decades of Web application testing—A survey of recent advances," Inf. Syst., vol. 43, pp. 20–54, Jul. 2014.

[11] P. Chen, N. Nikiforakis, L. Desmet, and C. Huygens, "Security Analysis of the Chinese Web," in Proceedings of the 2014 Workshop on Cyber Security Analytics, Intelligence and Automation - SafeConfig '14, 2014, pp. 3–9.

[12] Alexa, "Alexa - Top Sites in South Africa," 2015. [Online]. Available: http://www.alexa.com/topsites/countries;0/ZA. [Accessed: 13-Apr-2015].

[13] Mybroadband, "Top 20 South African websites," 2015. [Online]. Available: http://mybroadband.co.za/news/internet/116428-top-20-south-african-websites.html. [Accessed: 13-Apr-2015].

[14] T. van Goethem, P. Chen, N. Nikiforakis, L. Desmet, and W. Joosen, Large-Scale Security Analysis of the Web: Challenges and Findings, vol. 8564. Cham: Springer International Publishing, 2014.

[15] OWASP, "OWASP 2013 Top 10 Web Security Vulnerabilities," 2013. [Online]. Available: https://www.owasp.org/index.php/Top_10_2013-Top_10.

[16] OWASP, "List of useful HTTP headers - OWASP," 2014. [Online]. Available: https://www.owasp.org/index.php?title=List_of_useful_HTTP_headers&oldid=179254. [Accessed: 09-Apr-2015].

[17] L.-S. Huang, A. Moshchuk, H. J. Wang, S. Schecter, and C. Jackson, "Clickjacking: Attacks and Defenses," in USENIX Security Symposium, 2012, pp. 413–428.

[18] A. Sood and R. Enbody, “The state of HTTP declarative security in online banking websites,” Comput. Fraud Secur., vol. 2011, no. 7, pp. 11–16, Jul. 2011.

[19] N. Nikiforakis, Y. Younan, and W. Joosen, “HProxy: Client-side detection of SSL stripping attacks,” Detect. Intrusions Malware, …, 2010.

[20] A. K. Sood and R. J. Enbody, “Frametrapping the framebusting defence,” Netw. Secur., vol. 2011, no. 10, pp. 8–12, Oct. 2011.

[21] I. Dawson, “Guidelines for Setting Security Headers | Veracode,” 2014. [Online]. Available: https://www.veracode.com/blog/2014/03/guidelines-for-setting-security-headers. [Accessed: 14-Apr-2015].

[22] L. Constantin, “Low adoption rate of HSTS website sec,” PC World, 2014. [Online]. Available: http://www.pcworld.com/article/2140560/low-adoption-rate-of-hsts-website-security-mechanism-is-worrying-eff-says.html. [Accessed: 16-Apr-2015].

[23] D. Kristol and L. Montulli, “HTTP State Management Mechanism,” Internet Engineering Task Force, 1997. [Online]. Available: https://www.ietf.org/rfc/rfc2109.txt. [Accessed: 19-Apr-2015].