Analysis of Mozambican Websites: How do they protect their users?

Ambrósio Patrício Vumo[*][†], Josef Spillner[‡], Stefan Köpsell[†]
[*] Universidade Eduardo Mondlane, Maputo, Mozambique
[†] Technische Universität Dresden, Department of Computer Science, 01062 Dresden, Germany
[‡] Zurich University of Applied Sciences, School of Engineering, Service Prototyping Lab, 8401 Winterthur, Switzerland
Email: ambrosio_patricio.vumo@mailbox.tu-dresden.de, josef.spillner@zhaw.ch, stefan.koepsell@tu-dresden.de

***Abstract*—Web security is an important approach for most institutions, organizations and individuals which use or provide their services through websites. In this study, a systematic and methodical evaluation of the exposure of web servers and HTTP security headers to attackers that can cause potential harm was tested in 240 Mozambican websites. Vulnerabilities related to HTTP security headers were obtained and the mechanisms which should be taken to reduce the security risks of the services available on the websites are presented.**

***Index Terms*—Websites, Web services, Cryptographic protocols, Network security, HTTPS, TLS, Vulnerability**

I. Introduction

Web security is concerned with the technological protection of all individual participants, as well as consumer organizations and service providers, on simple websites and in complex web applications [1]. The concern of web security is among the everlasting unsolved issues on the Internet suppressing wider deployment of web-hosted applications for critical tasks. Increasingly, it is also the part of the Internet that is most vulnerable to attacks [1]. Initially, the emphasis in this domain focused on the problem of protecting data and information represented by it transmitted over web protocols from web servers to the end users without revealing it to unauthorized third parties. According to studies [2], the number of companies and individuals with Internet access is expanding rapidly. As a result, most institutions and organizations are actively involved in setting up facilities on the web for providing services. Worse from a consumer choice point of view, often the use of these services becomes mandatory for employees, customers or citizens, often in combination with mobile clients. Examples include ticketing systems, tax filing and cloud document management. Yet at the same time, websites and services embedded therein are often vulnerable to attacks on the data transmission beyond unavailability and other distortions. Hence, the demand for secure websites continues to grow and becomes of interest to all political levels.

In recent years the government of Mozambique has been emphasizing the integration of ICT in order to improve communication and the exchange of information in public and private institutions [3]. According to ITU [4] through a ”Cyberwellness Profile”, in 2016, the Mozambique parliament have approved the first legislation on electronic transaction bill that intends to regulate the use of electronic systems in trade, finance and other areas. The legislation also intends to protect consumers against cybercrime and electronic frauds.

The goal of this study is to make an analysis of the efforts of various institutions in implementing security mechanisms in their websites in the light of this development. Thus, we have analyzed web server security options, HyperText Transport Protocol (HTTP) security headers used in Mozambican websites and the mechanisms that should be taken to reduce security risks of the websites. The purpose of this analysis is to evaluate how many Mozambican institutions use proper security mechanisms to protect their consumers while serving them. On the other hand, we call attention to the importance of using the security mechanisms and the risks that can be caused when some mechanisms are not observed. The paper is divided into five sections. Section II describes the research approach and how the data were collected. Section III reviews HTTP security headers and how the web server can be secured. In Section IV we present discussion of results obtained and related work. Finally, Section V presents conclusions and recommendations to all operators of websites in Mozambique.

II. Research methodology and data collection

In this section, we discuss the methodology adopted for this study and the techniques used to identify websites in Mozambique and for collecting data from the selected set of websites.

A. Research approach

To achieve our goal, a methodology proposed by [5] was adapted. This approach is based on vulnerability assessment which is the evaluation process of finding, numbering and ordering vulnerabilities or threats to a system. The assessment determines the exposure of assets to active attackers, but also higher forces such as natural disasters which could negatively interfere with the provided services. It assists in determining the need for protection (asset identification), the degree of protection against the pressure exploiting the vulnerability (threat evaluation), the quality of protection mechanisms currently in place (vulnerability appraisal), and a risk analysis taking the potential damage into account (risk assessment). The outcome of any such analysis is a plan on what to change to avoid the risks (risk mitigation). There are two different vulnerability assessment techniques [5]: i) vulnerability scanning and, ii) penetration testing (pentesting). The scanning technique is an automated software search (scan) through a system, in our scope a web server, for known weaknesses. The scan creates a report of the potential vulnerabilities. It thus examines the currently achievable level of security through a passive method of testing security controls and is typically performed with full access privileges from inside the system. Depending on how it is performed, it may execute alongside the day-to-day operations without interruption. In contrast, penetration testing is more intrusive and may exploit actual weaknesses in vulnerable systems. It cannot be fully automated as the skills, technical capabilities and malicious creativity of the person running the test are crucial to the success. Compared to the scanning approach, pentesters are often located outside of systems, for instance as hired consultants. Their actions can furthermore disrupt the system operations and cause irreversible damage which calls for a controlled approach.

Passive vulnerability assessment (i.e. scanning) was adapted as main methodology and the desk research methodology was adapted through online desk research technique as second methodology to collect published reports [6]. Furthermore, the chosen research approach includes a choice of processing the results. In vulnerability assessment, there are two techniques to choose from: i) Baseline Reporting and ii) Software Program Development [5]. In baseline reports, the current state of a system is compared to a well-defined baseline. The value of the reports is the ability to quickly discover unusual behaviour in systems which may indicate an attack or a new vulnerability. In software program development, flaws are minimized through a security-conscious software development which leads to secure software being released instead of security patching after the release. For this study, baseline report was adapted as assessment technique. The reports are consequently made available for verification as integral part of the research output.

B. *Selected tools and data collection*

The identification of vulnerabilities can be performed with many different tools. Most of them can be used both by security analysts and by attackers, rendering them into dual-use software [5]. The dominant tool categories are: i) Port scanners, ii) Banner grabbing tools, iii) Protocol analyzers, iv) Vulnerability scanners, and v) Honeypots. Vulnerability scanners were adapted in this study and the following tools were used for data collection:

- **Nmap:** Nmap systematically connects to well-known ports on all detected hosts on a network with the purpose to collect information about the offered services. The host detection uses raw Internet protocol packets. It then parses service fingerprints, compares them to a database and is thus able to identify software programs and their versions used to implement the services as well as operating systems and packet filters deployed on each host [7].
- **sslscan:** SSL/TLS-protected services, such as HTTPS, are examined by sslscan to determine the supported cryptographic ciphers as well as the certificates in use [8]. The tool stands out by being lean and fast.
- **Nikto:** Nikto examines the state of a web server. It reports on expected default files, including known insecure ones, as well as the software programs and configurations on each checked server [9]. Thus, in addition to the previous tools, potential site-specific security vulnerabilities can be detected which include outdated programs and insecure configurations.

Table I
THE WEBSITES ANALYZED

Nr.	Category	Websites
1	Telecommunications	7
2	Government	33
3	Academic	19
4	Bank	16
5	Media	15
6	Company	129
7	Polical party	3
8	Other organization	17
	Total	**240**

C. *Website selection*

In order to identify and collect data from websites for analysis, the Woorankindex site was used [10], which provides global statistics about the Internet, such as: top sites, web servers technical information and local top sites. Also, the National Government portal was analyzed [11]. The data has been collecting in February and March 2017. Thus, 240 websites were selected as Mozambique top sites in accordance with Woorankindex. The websites were organized in 8 categories: telecommunications, government, academic, banks, media, companies, political parties and other organizations (see Table I).

III. BACKGROUND: HTTP SECURITY HEADERS AND SECURING THE WEB SERVER

This section will focus on the first three potential attack points, i.e, the machine used to run the client components, server-side components and the communication that takes place between client and server side.

A. *HTTP security headers*

The 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 [12]. We take a look at the HTTP security headers as recommended by the Open Web Application Security Project

(OWASP) [13]. The following headers can be utilised to increase the security of websites [14], [15]:

- **HTTP cookies:** Cookies are small amounts of data stored on the device of the user by the web browser on request of the web server depending on the configured browser policy. They are therefore commonly referred to as web or browser cookies although the more precise term is HTTP cookies. The function of cookies is to orrelate subsequent requests from the same user to a website or web application, allowing a user to stay logged in or to maintain the shopping basket on multiple web pages, for instance. Thus, the use of cookies, specified in RFC 2109 and more recently in RFC 6265, leads to stateful HTTP sessions with mainly three purposes: session management (logins, carts), personalization (persistent user preferences on a website) and user tracking [16]. The secure use of cookies is therefore paramount to the overall website security, and how to use cookies correctly is still being debated in web and security communities [17]. Two distinct headers exist for this purpose. **Secure Cookie**: This attribute tells browser to send the cookie to a server only over HTTPS connections. If not set, the cookie will be submitted over any type of connection as the browser does not know that the cookie is sensitive and in need of protection [16]. **HttpOnly Cookie**: Prevents access to sessions through Cross-Site Scripting (XSS) exploits which is one of the most common attack vectors. Successful attacks are typically followed by a subsequent hijacking of the victim's session. The HttpOnly header therefore mitigates the attack by preventing access to cookie values through client-side JavaScript execution.
- **X-Frame-Options:** The header is designed to mitigate clickjacking attacks in which users unconsciously follow links to malicious content. To stop clickjacking, it instructs the browser whether a webpage is allowed to be embedded in a frame or other object on the page. Websites can use the header to ensure that foreign content is not embedded into ther websites. There are three possible directives for X-Frame-Options: DENY, SAMEORIGIN and ALLOW-FROM [url].
- **X-XSS-Protection:** The header is a vendor-specific feature of older web browsers which prevents pages from loading when they recognize XSS attacks, nowadays largely superseded by Content Security Policy. There are four options for X-XSS-Protection: Disable XSS filtering (0), enable XSS filtering (1) in case an XSS attack is detected, the page will be sanitized by removing the unsafe parts, enable XSS filtering (1;mode=block) and enable XSS filtering (1;report=[reporting-URI]) in case an XSS attack is detected, the violation will be reported on top of the unsafe parts removal.
- **Content Security Policy:** As successor to X-XSS-Protection standardized by the W3C, detects and mitigates attacks involving XSS and data injections such as unauthorized access to data, website defacement or distribution of maleware. This header is parameterized with values to restrict the browser in loading additional resources for any page such as scripts, images, fonts or media. It can also restrict form submission targets.
- **X-Content-Type-Options:** This header was introduced by Microsoft in Internet Explore 8 as a way for webmasters to block content sniffing that was happening and could transform non-executable MIME types into executable MIME types. There is one value for X-Content-Type-Options: nosniff, it blocks a request if the requested type is ”style” and the MIME type is not ”text/css” or ”script” and the MIME type is not a JavaScript MIME type.
- **Strict-Transport-Security:** This response header (often abbreviated as HSTS) is a feature that lets a website tell browsers that it should only be communicated with using HTTPS. The effect is that HTTP accesses are forcibly upgraded to HTTPS. There are three possible directives for HSTS: max-age=[expire-time], includeSubDomains and preload. Thus, the HSTS header is ignored by the browser when a website is accessed using HTTP; this is because an attacker may intercept HTTP connections and inject the header or remove it. When a website is properly accessed over HTTPS, the browser knows that the website is HTTPS capable and will honor the HSTS header. This policy protects against two categories of attacks: passive eavesdropping and active transmission hijacking through Man-in-the-Middle (MITM) attacks.

B. Securing the web server

As the aim of the research is to convey practical recommendations to website operators, the procedures and technologies for securing the servers and services will be briefly presented to ultimately protect consumers and providers of web applications alike [1]. HTTP was designed in 1991 to serve HyperText Markup Language (HTML) over the Internet [18]. However, it evolved since then to support much more than just static HTML pages. HTTP is used to communicate between browser clients and web servers, and care must be taken when considering the data going across the network, as with plain HTTP, data is not encrypted in any way and passed across the connection in a human-readable format. To encrypt the data, its secured variant HTTPS was developed which is typically served on port 443 instead of 80 for plain HTTP. Yet, merely offering HTTPS is not sufficient to increase security. According to [6], the encryption when using HTTPS is handled transparently, with the client and server first communicating the TLS handshake. This allows the protocol to pass data between the two tiers that will be used for the encryption process, and the most important piece of data is the server's X.509 certificate. Certificates are digital files that contain information about the server machine, and most importantly contain the server's public key.

HTTPS is designed to provide strong security. Yet it may fall short of the specified goals when it is incorrectly implemented in server-side components. A number of security issues associated with this protocol have been unwrapped over time, starting from cryptographic limitations and design suboptimal choices in the TLS protocol, to the insecure deployment and configuration of HTTPS websites. To configure a secure HTTPS web server, there are a number of security glitches to be avoided. The web server administrator must observe the following security issues:

- **Validity period of digital certificate:** In accordance with [19], many Certificate Authorities (CAs) have decided in 2015 to stop the creation of TLS certificates whose validity interval exceeds 39 months [19], [20]. The rationale for this change is the sweetspot between high usability and high connection security. In contrast to previous certificates with longer intervals, administrators are now required to review and update their certificates more frequently, thus heuristically minimizing the attack surface for known vulnerabilities. Security-conscious websites will most likely aim for even shorter intervals. Wikileaks certificates, for instance, have a validity of three months, but in turn require more frequent administrator action. A remaining issue is that even browser warnings about outdated certificates are often ignored by users due to trained indifference [21].
- **Trust in CAs:** According to [22], trusted CAs are organizations or companies which issue signed digital certificates, often for a fee. The responsibilities of these organizations include identity and background checks on websites and their owners for which they emit certificates, setting them apart from unsigned or self-signed ones. A self-signed certificate, in contrast, would be signed by the website operators themselves. Self-signed certificates are not recommended for use by websites because they do not follow the best practices, may not follow industry guidelines and have not been audited Yet apart from commercial CAs [19], community CAs such as CAcert and Let's Encrypt continue to increase market share.
- **Algorithms:** Hash functions are used in many important security-critical applications like digital signatures, timestamps, message authentication codes and authentication protocols [23]. Attacks against hash functions may thereby have a large influence on the overall security of electronic services. Several hash functions such as MD5 and SHA1 [23]–[26] that are still in use in some applications today have been successfully attacked in terms of collisions. A collision attack is an attempt to find two input strings of a hash function that produce the same hash result.

C. Implemention of HTTP security headers

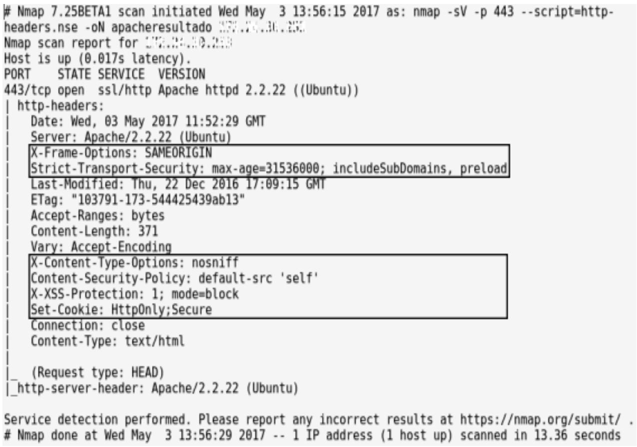
This subsection exemplarily provides prescriptive guidance for establishing a secure configuration posture for the web servers Apache and Nginx [27], [28] through HTTP security headers to demonstrate the relative ease of achieving secure websites. The instructions were tested on Apache httpd 2.2.22 in Ubuntu 12.04.5 and Nginx 1.4.6 in Ubuntu 14.04.5 LTS. They provide proper mitigation measures to either remove weaknesses or reduce the risks level. Security vulnerabilities related to HTTP headers can be fixed by implementing the necessary configuration in the file httpd.conf located in /etc/apache2/ for Apache and nginx.conf located in /etc/nginx/ for Nginx. Thus, for httpd.conf the Header directive was used under module [IfModule mod_headers.c][/IfModule] while for nginx.conf the add_header directive was used under the http block. These configuration can also be applied per-site inside the respective virtual host configurations.Table II shows how to configure HTTP security headers.

Table II
THE HTTP HEADERS CONFIGURATION

Headers	**Web Server**	**Setting**
HTTP Cookies	Apache	Header set Set-Cookie HttpOnly;Secure
	Nginx	add_header Set-Cookie "HttpOnly;Secure";
X-Frame-Options	Apache	Header always append X-Frame-Options SAMEORIGIN
	Nginx	add_header X-Frame-Options "SAMEORIGIN";
X-XSS-Protection	Apache	Header set X-XSS-Protection "1; mode=block"
	Nginx	add_header X-XSS-Protection "1; mode=block";
X-Content-Type-Options	Apache	Header set X-Content-Type-Options nosniff
	Nginx	add_header X-Content-Type-Options nosniff;
Strict-Transport-Security	Apache	Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
	Nginx	add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload";
Content Security Policy	Apache	Header set Content-Security-Policy "default-src 'self'"
	Nginx	add_header Content-Security-Policy "default-src 'self'";

D. Testing HTTP security headers

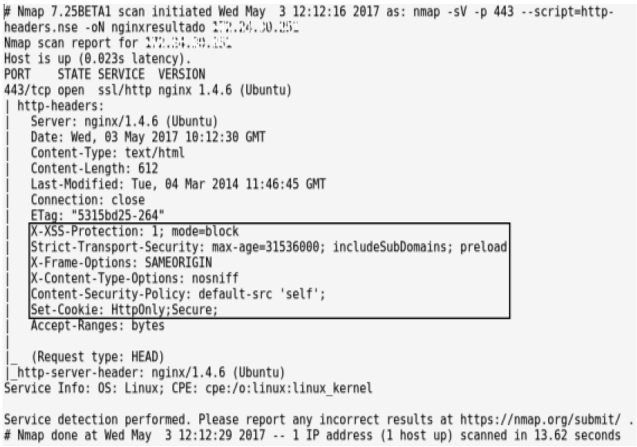
The technique to test the configuration is the same we apply to investigate the state of website security in Mozambique. In this subsection we describe the tests made in our laboratory environment and results of HTTP headers. For this test, the



```
# Nmap 7.25BETA1 scan initiated Wed May  3 13:56:15 2017 as: nmap -sV -p 443 --script=http-
headers.nse -oN apacheresultado [redacted]
Nmap scan report for [redacted]
Host is up (0.017s latency).
PORT    STATE SERVICE  VERSION
443/tcp open  ssl/http Apache httpd 2.2.22 ((Ubuntu))
| http-headers:
|   Date: Wed, 03 May 2017 11:52:29 GMT
|   Server: Apache/2.2.22 (Ubuntu)
|   X-Frame-Options: SAMEORIGIN
|   Strict-Transport-Security: max-age=31536000; includeSubDomains, preload
|   Last-Modified: Thu, 22 Dec 2016 17:09:15 GMT
|   ETag: "103791-173-544425439ab13"
|   Accept-Ranges: bytes
|   Content-Length: 371
|   Vary: Accept-Encoding
|   X-Content-Type-Options: nosniff
|   Content-Security-Policy: default-src 'self'
|   X-XSS-Protection: 1; mode=block
|   Set-Cookie: HttpOnly;Secure
|   Connection: close
|   Content-Type: text/html
|
|_  (Request type: HEAD)
|_http-server-header: Apache/2.2.22 (Ubuntu)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed May  3 13:56:29 2017 -- 1 IP address (1 host up) scanned in 13.36 seconds
```

Figure 1. The HTTP security headers on Apache



```
# Nmap 7.25BETA1 scan initiated Wed May  3 12:12:16 2017 as: nmap -sV -p 443 --script=http-
headers.nse -oN nginxresultado [redacted]
Nmap scan report for [redacted]
Host is up (0.023s latency).
PORT    STATE SERVICE  VERSION
443/tcp open  ssl/http nginx 1.4.6 (Ubuntu)
| http-headers:
|   Server: nginx/1.4.6 (Ubuntu)
|   Date: Wed, 03 May 2017 10:12:30 GMT
|   Content-Type: text/html
|   Content-Length: 612
|   Last-Modified: Tue, 04 Mar 2014 11:46:45 GMT
|   Connection: close
|   ETag: "5315bd25-264"
|   X-XSS-Protection: 1; mode=block
|   Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
|   X-Frame-Options: SAMEORIGIN
|   X-Content-Type-Options: nosniff
|   Content-Security-Policy: default-src 'self';
|   Set-Cookie: HttpOnly;Secure;
|   Accept-Ranges: bytes
|
|_  (Request type: HEAD)
|_http-server-header: nginx/1.4.6 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed May  3 12:12:29 2017 -- 1 IP address (1 host up) scanned in 13.62 seconds
```

Figure 2. The HTTP security headers on Nginx

Nmap tool and a custom nmap script (http-headers.nse) were used.

- **Testing Apache:** Figure 1 presents results obtained from a local Apache web server. The results indicate the HTTP security headers configured on the server, i.e, Secure Cookie, HttpOnly Cookie, X-Frame-Options, X-XSS-Protection, Content Security Policy, X-Content-Type-Options and Strict-Transport-Security.
- **Testing Nginx:** Figure 2 presents the results obtained from Nginx. The result are equivalent. Thus, both web servers are adequately configured.

IV. Results and discussion

We present results of our evaluation of 240 tested websites in Mozambique based on the HTTP security headers and mechanisms used to secure the web server.

A. Results

A total of 240 websites in the ”mz” domain were analyzed and eight groups were created: i) The Telecommunications category is constituted of Internet Service Providers and Mobile Operators, ii) The Government category is constituted by government websites at different levels, iii) The Academic category is the set of higher education institution and universities, iv) Media, this category is constituted by newspaper and television channel websites, v) Bank, this category presents the banks, vi) Political party, set of political parties and viii) Other organizations, for the remainder. The study is based on the following metrics: Cookie attributes (secure and HttpOnly), X-Frame-Options, X-XSS-Protection, X-Content-Type-Options, HSTS, HTTP and HTTPS implementation, trusted certificate, self-signed certificate, validity period of digital certificate, CA and signature algorithms.

- **HTTPS implementation:** As can be seen in Table III, only 76 websites which represent 32% of 240 of ”mz” domain websites evaluated are not using HTTPS whereas 164 websites which represent 68% use HTTPS, 15 of them exclusively.

Table III
THE WEBSITES USING HTTP/HTTPS

Nr.	Category	HTTP Only	HTTPS Only	Both (HTTP and HTTPS)
1	Telecommunications	3	0	4
2	Government	11	3	20
3	Academic	3	2	14
4	Bank	8	2	6
5	Media	3	0	12
6	Company	44	7	78
7	Polical party	0	0	3
8	Other organization	4	1	12
	Total	**76**	**15**	**149**

- **Trusted certificate authority and self-signed certificate per group:** Figure 3 shows the percentage of websites which implement HTTPS per category and per CA. Our results indicate 83% of the government websites use self-signed certificates and only 17% use trusted CAs. This statistic is based on the 164 websites using HTTPS.
- **Trusted certificate authority and self-signed certificate:** In total (Figure 4), 78% of the websites are using trusted CAs and 22% are using self-signed certificates.
- **HTTP Strict-Transport-Security:** Figure 5 indicates that 2% of all websites with HTTPS have configured HSTS, i.e, only 3 websites have implemented Strict-Transport-Security.
- **X-Content-Type-Options, Content Security Policy and X-Frame-Options:** We see also on Figure 5 that 0% of all websites analyzed which had implemented HTTPS were configured to make use of X-Content-Type-Options. The same holds for X-XSS-Protection and X-Frame-Options, as well as the two cookie-related headers

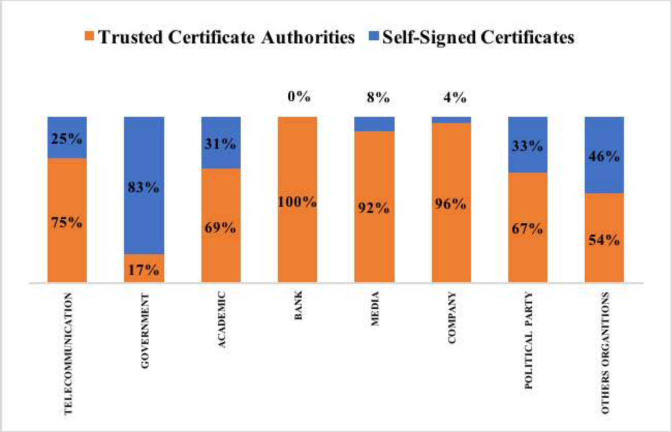


Figure 3. Websites using trusted CAs and self-signed certificates, across categories

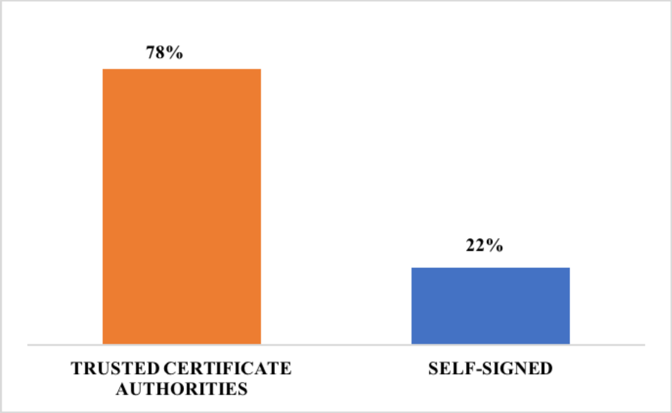


Figure 4. Websites using trusted CA and self-signed certificate, national average

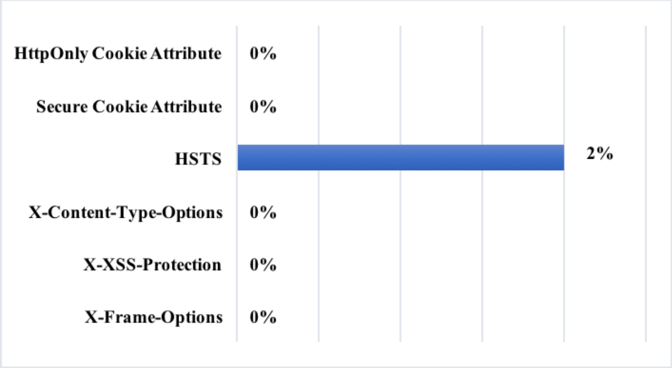


Figure 5. HTTP security headers

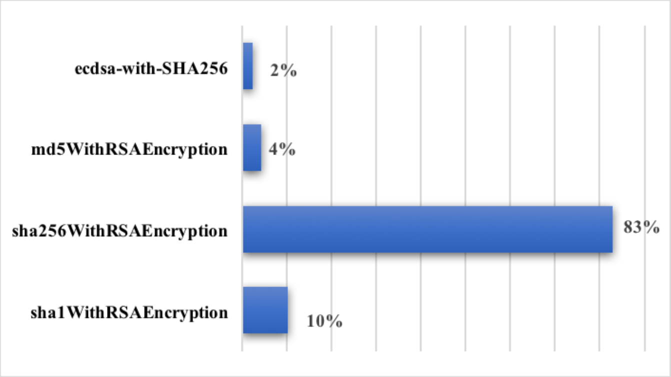


Figure 6. Hash algorithms used in websites

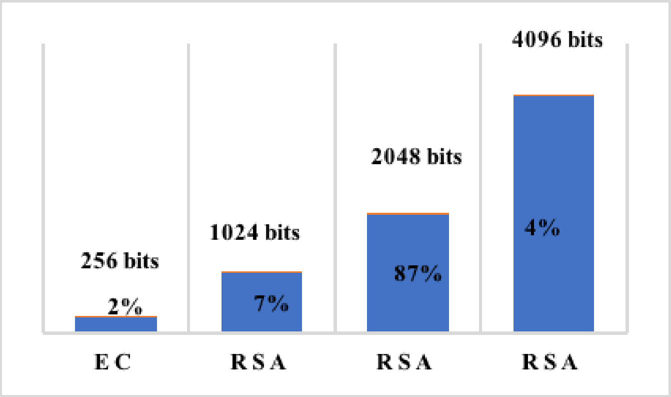


Figure 7. Public key length

Table IV
VALIDITY OF CERTIFICATES

Expired (2012 to 2016)	Expired (Jan to Mar 2017)	Validity (Apr to Dec 2017)	Validity (2018 to 2046)
15 %	6 %	42 %	37 %

HttpOnly and Secure Cookie. This surprising result indicates that more recent security measures are not yet known to administrators of these websites, or are not required for the type of website.

- **Hash Algorithm:** Figure 6 shows that old and considered unsafe hash algorithms [23]–[26] like MD5 and SHA1 are still used. 4% of all certificates use MD5, and 10% of all certificates use SHA1.
- **Public key length (in bits):** Figure 7 depicts the distribution of public key length bits and public key type (RSA and EC algorithm). Again, there is a need to secure. 7% of RSA algorithms work with 1024 bits which are widely considered insecure [20], [24]–[26], [29].
- **Validity of digital certificates:** As Table IV shows, 15% of certificates analyzed had expired between 2012 and 2016, and an additional 6% had expired between January and March 2017, the time of the study. In contrast, 42% have their remaining validity between April and December 2017 and 37% between 2018 and 2046.
- **Certificate validity period:** Independent of the time of expiration, another interesting metric is the overall validity of a certificate. It can be noted on Table V that

Table V
CERTIFICATES VALIDITY

Period (1 to 3 Years)	Period (3 to 10 Years)	Period (10 to 40 Years)
88 %	10 %	2 %

88% of the websites follow the best practices, i.e, digital certificates have a validity not greater than 3 years and 12% of websites do not follow the best practices by offering certificates which could be trusted by clients long after the relevant ciphers may become broken [19], [20].

B. *Data discussion*

The results of this research show the following HTTP security headers were related to problems flagged by tools used in this study. Thus, four vulnerabilities were obtained, namely: the headers X-XSS-Protection, X-Frame-Options, Strict-Transport-Security and X-Content-Type-Options which were missing in configurations of web servers. This means that most web servers have been configured with default configuration, the security misconfiguration and lack of knowledge about common vulnerabilities. In relation to websites which use certificate authorities, the results indicate that roughly four out of five use trusted CAs, yet among government websites the ratio is reversed. This could signal a relative low priority of website security in authorities compared to companies where much more is at stake, including reputation and business transactions. In relation to maximum certificate validity our results obtained show that 12% of websites do not follow the best practices especially concerning digital certificates, i.e, the digital certificates have a validity greater than 3 years. This result indicates that the websites are not audited or the responsible administrators of these websites have a lack of knowledge of the consequences of certificates with long periods of validity. It should be noted that the sample size is rather small and may therefore not be representative, although the national registrar, the Center of Informatics of Eduardo Mondlane University (CIUEM) responsible for most ”mz” domains, does not release any numbers. Future work will include frequent rescans to track the results and possible improvements over time. Furthermore, potentially all Mozambican public IPs obtained through the GeoIP2 country database will be evaluated, leading to a much greater sample size.

C. *Related work*

There are several researches that have been conducted such as: J. Mtsweni [30] in South Africa with his study titled ”Analyzing the Security Posture of south African Websites”, Ping Chen et al [31] in China with their study titled ”Security Analysis of the Chinese Web: How well is it protected” and Jeremy Clark, C. van Oorschot [32] in Canada with their study titled ”SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements”. In these researches we evaluated the following cases: the security of websites based in HTTPS implementation, HTTP security headers and evaluating certificate trust model as related works. Furthermore presenting the approaches presented in the related works, this study was conducted by following all phases proposed by [5], in this context, after identifying the risks in websites, we present practical solution, i.e, how to mitigate these risks by presenting the configurations made in the web servers (Apache and Nginx) as shown in section III through subsection C and D. Also according to the desk research methodology through online desk research technique there are no similar studies about this approach in Mozambique. According to [30], most of the 70 South African websites are vulnerable to common attacks (such as clickjacking and cross-site scripting) but judging by the Mozambican websites studied by us the data shows that more than 160 websites, or around two thirds, are vulnerable to common attacks, which calls for a more urgent and prioritised tacking of the issue.

V. CONCLUSION AND RECOMMENDATION

This section presents conclusions and recommendations for government through Mozambique National Institute of Communication (INCM), National Institute of Technologies and Communication (INTIC) and all owners of websites in Mozambique.

A. *Conclusion*

The purpose of this analysis was to evaluate how many Mozambican institutions use the security mechanisms available from the web server by implementing HTTP(S) security headers to provide their services and protect their consumers. Thus, the following conclusions were obtained:

- None of the websites is configured to use the following HTTP security headers: X-Content-Type-Options, X-XSS-Protection, X-Frame-Options, Secure Cookies and HttpOnly Cookies;
- Most government’s websites analyzed are using self-signed certificates without centrally enforced management policies and only four government websites use trusted certificates authorities;
- The results indicate that 2% of websites analyzed which had implemented HTTPS had configured HTTP Strict-Transport-Security. This represents only 3 of 164 websites;
- Most self-signed certificates used in websites either have a maximum certificate validity of greater than 3 years or with an expired date, leading to insecurity in both cases.

Those results demonstrate a lack of: security monitoring, national entities responsible for security monitoring and knowledge about web server security and vulnerabilities, and above all a coherent national strategy related to digital services including aspects of security.

B. Recommendation

We would like to end this study with recommendations for mitigating and deterring attacks through security posture and security strategy. Thus, the following recommendations should be considered by all owners of websites:

- Regular monitoring of systems and networks through vulnerability assessment tools to provide valuable information regarding the current state of security and their mitigation;
- Good security requires having a secure configuration defined and deployed for applications, frameworks, application servers, web and database servers, and platforms. Secure settings should be defined, implemented, and maintained, because defaults are often insecure [13];
- It is necessary to have regulation to enforce all private and public organizations implementing security mechanisms to lower security risks, for the greater benefit of the society and its future development.
- Thus, looking at future requirements, we recommend the establishment of a lean regulatory entity which defines a strategy for website and web applications/cloud applications security and regularly assesses the state of government, business, academic and private websites to achieve a world-renowned state of secure digital information and services.

C. Material

We encourage the confirmability of our claims and the repeatibility of our work by sharing the technical underpinnings of the research presented in this paper. The raw material including the list of websites and their use of HTTP headers is curated at an Open Science Framework repository at `https://osf.io/35sz8/`.

References

[1] Garfinkel S. and Spafford G., *Web Security, Privacy and Commerce*, 2nd ed. Sebastopol: O'Reily and Associates, 2002.

[2] Stallings W, *Cryptography and Network Security: Principles and Practice*, 5th ed. Prentice Hall, 2005.

[3] Comparative Report 2005, C. Alberto, S. Manhica, Z. Saifodine and A. Nunes Junior, "e-Government, eHealth, Technology Enhanced Learning: Adoption in Mozambique, South Africa and Tanzania," 2005.

[4] ITU. (2017, Mar.), "Cyberwellness profile: Republic of Mozambique." [Online]. Available: https://www.itu.int/en/

[5] Ciampa M., *Guide to Network Security Fundamentals*, 5th ed. Boston: Cengage, 2015.

[6] MSG Management. (2017, Feb.), "Desk Research – Methodologies and Techniques." [Online]. Available: http://www.managementstudyguide.com/desk-research.htm

[7] Nmap. (2017, Feb.) , "Web server vulnerability scanner." [Online]. Available: https://nmap.org/

[8] Sslscan. (2017, Feb.), "SSL scanner." [Online]. Available: http://sslscan.sourcearchive.com/documentation/1.7.1-1/main.html

[9] NIKTO. (2017, Feb.), "Web server vulnerability scanner." [Online]. Available: https://cirt.net/nikto2

[10] woorankindex. (2017, Feb.), "WooRank Reviews Index: TOP 317 Websites in Mozambique." [Online]. Available: https://index.woorank.com/en/reviews?country=mz

[11] INTIC. (2017, Mar.), "Mozambican government portal." [Online]. Available: http://www.portaldogoverno.gov.mz

[12] A. Sood and R. Enbody, "The state of HTTP declarative security in online banking websites," 2011.

[13] WASP. (2017, Feb.), "Web Application Security Project." [Online]. Available: https://www.owasp.org

[14] Mozila Developer Network. (2017, Feb.), "HTTP headers." [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

[15] Michael Skiba. (2017, Feb.), "The Security of HTTP-Headers." [Online]. Available: https://www.contextis.com/resources/blog/security-http-headers

[16] A. Barth, "HTTP State Management Mechanism," *IETF RFC 6265*, 2011.

[17] P. Rabinovich, "Secure cross-domain cookies for HTTP," *Journal of Internet Services and Applications*, vol. 4, no. 13, 2013.

[18] David G. and Brian T., *HTTP: The Definitive Guide*, 1st ed. Gravenstein Highway North: O'Reilly Media, 2002.

[19] CA/Browser Forum. (2017, Mar.), "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates," pp. 13–14. [Online]. Available: https://cabforum.org/wp-content/uploads/BRv1.2.3.pdf

[20] Entrust Datacard. (2017, Mar.), "SSL/TLS Best Practices." [Online]. Available: https://www.entrustdatacard.com/wp-content/uploads/Entrust-eGuide-SSL-Best-Practices-V3-WEB.pdf

[21] A. Bostan, "Implicit learning with certificate warning messages on ssl web pages: what are they teaching?" *Security and Communication Networks*, vol. 9, no. 17, pp. 4295–4300, 2016, sCN-15-0666.R1. [Online]. Available: http://dx.doi.org/10.1002/sec.1607

[22] Zakir Durumeric, et al, "Analysis of the HTTPS Certificate Ecosystem," IMC Conference. Proceedings of the 2013 conference on Internet measurement conference, 2013.

[23] Estonian Information System Authority. (2017, Mar.), "Cryptographic algorithms lifecycle report 2016." [Online]. Available: https://www.ria.ee/public/RIA/

[24] ECRYPT. (2017, Mar.), "European Network of Excellence in Cryptology II." [Online]. Available: http://www.ecrypt.eu.org/ecrypt2/documents/D.SPA.20.pdf

[25] NIST. (2017, Mar.), "Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths." [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf

[26] David A. Wheeler., *Secure Programming for Linux and Unix HOWTO*, 3rd ed. Boston: Free Software Foundation, 2003.

[27] Members of the Ubuntu. (2017, Mar.), "Ubuntu Server Guide." [Online]. Available: https://help.ubuntu.com/lts/serverguide/serverguide.pdf

[28] Nginx. (2017, Mar.), "Configuring HTTPS Servers." [Online]. Available: https://nginx.org

[29] BlueKrypt. (2017, Mar.), "Cryptographic Key Length Recommendation." [Online]. Available: https://www.keylength.com/en/compare/

[30] J. Mtsweni, "Analyzing the Security Posture of South African Websites," March 2015.

[31] P. Chen, N. Nikiforakis, L. Desmet and C. Huygens, "Security Analysis of the Chinese Web: How well is it protected," 2014.

[32] J. Clark and P.C. van Oorschot, "SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements," *2013 IEEE Symposium on Security and Privacy*, 2013.