Using Soft Systems Methodology and Activity Theory to Exploit Security of Web Applications against Heartbleed Vulnerability

Maduhu Mshangi4, Edephonce Ngemera Nfuka, [3]Camilius Sanga
[1]NECTA – Tanzania, [2]Open University of Tanzania, [3]Sokoine University of Agriculture
[*]Corresponding author e-mail: mshangimaduhu@yahoo.com

ABSTRACT

The number of security incidents exploiting security holes in the web applications is increasing. One of the recently identified vulnerability in the web applications is the Heartbleed bug. The Heartbleed bug is a weakness found in OpenSSL, open source cryptographic software. In this study, both quantitative and qualitative research methodologies were employed. Case study and content/documentary analysis research methods were used to collect data for probing the web applications which are vulnerable to the bug. Due to the complexity of the problem, Soft Systems Methodology was adopted for the management of the analysis of data. The evaluation of security of web applications involved 64 selected websites of higher education institutions in Africa. SSM was supported by a theory called Activity Theory. The collected data was analysed using "R statistical computing package". The study found that 89% of the universities web applications in Africa were vulnerable to the Heartbleed attack; and 11% of the universities web applications in Africa were not vulnerable to Heartbleed on the public announcement of the bug. But about two months later after the public announcement of the bug, 16% of the most universities web applications which were vulnerable were patched for the Heartbleed bug. The study seeks to contribute in application of Soft Systems Methodology and Activity Theory in the body of knowledge of information systems security (ISS).

Categories and Subject Descriptors: K.6.1 [**Management of Computing and Information Systems**] Project and People Management.

General Terms: Management
Additional Key Words and Phrases: Heartbleed bug, web application, security, SSL, TLS, SSM, Activity theory

IJCIR Reference Format:

Maduhu Mshangi, Edephonce Ngemera Nfuka, and Camilius Sanga. Using Soft Systems Methodology and Activity Theory to Exploit Security of Web Applications against Heartbleed Vulnerability. International Journal of Computing and ICT Research, Vol. 8, Issue 2 pp 32-52. http://ijcir.mak.ac.ug/volume8-issue2/article4.pdf

4 Authors Address: Maduhu Mshangi, 2Edephonce Ngemera Nfuka, [3]Camilius Sanga, [1]NECTA – Tanzania, [2]Open University of Tanzania, [3]Sokoine University of Agriculture, [*]Corresponding author e-mail: mshangimaduhu@yahoo.com. International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online), Vol. 8, Issue 2, pp 32-52, December 2014

1. INTRODUCTION

Internet is the foremost leading media for information exchange and doing business today (Kumar and Gade, 2011). As the world embraces the Internet and cloud computing, more and more people are transacting business, conducting research, storing information, collaborating with co-workers, publishing personal thoughts, and fostering relationships via web applications. Each time you launch a browser and connect to a website, you're using one or more web applications. With web applications, the bulk of processing occurs on servers located at remote sites. As a result, users can run sophisticated web applications from virtually any PC, a low-powered notebook, a tablet computing device, or smartphone. Web applications are generally easy to use, are efficient, and pervasive. This is why web applications have become the Achilles' heel of Information Technology (IT) security (Shema, 2011).

The exploitation of web applications in the cyberspace is the global security concern. The number of security incidents exploiting vulnerabilities in web applications is rising (IPA, 2013; CVE, 2014). Recently, web applications security incidents have become for-profit and are getting more vicious. More than 7,412 web applications vulnerabilities have been reported to Information-Technology Promotion Agency (IPA) (IPA, 2013, pp.4-5). The Common Vulnerabilities Exposures (CVE) database has registered more than 13,495 web applications vulnerabilities (CVE, 2014) up to July 30, 2014; the most common web applications vulnerabilities are namely: SQL Injection, Operating System (OS) Command Injection, Unchecked Path Parameter / Directory Traversal, Improper Session Management, Cross-Site Scripting, Cross-Site Request Forgery (CSRF), HTTP Header Injection, Mail Header Injection, Lack of Authentication and Authorization (IPA, 2011). Despite the mentioned vulnerabilities, recently the web application vulnerability called Heartbleed bug has been disclosed.

The Heartbleed bug is vulnerability in the OpenSSL cryptographic software library (Heartbleed.com, 2014; OpenSSL, 2014). It allows stealing the information protected, under normal conditions, by the Secure Sockets Layer / Transport Layer Security (SSL/TLS) encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs) (OpenSSL, 2014). The Heartbleed bug was introduced to OpenSSL on December 2011 and has been out in the wild since OpenSSL released 1.0.1 on 14th of March 2012; OpenSSL 1.0.1 through 1.0.1f, as well 1.0.2-beta. The vulnerability allows a hacker to access the memory of servers where sensitive information can reside. This vulnerability can potentially impact Internet communications and transmissions. The vulnerability has existed since 2012, but was only recently publically announced on April 07, 2014; thus cyber-criminals could exploit this vulnerability to intercept and decrypt encrypted information (Heartbleed.com, 2014).

1.1. Problem Statement

Vulnerability in OpenSSL could allow an attacker to expose memory contents from a server, which could result in exposure of confidential information such as usernames, passwords, session IDs, and secret keys used for encryption. Exploit code for the OpenSSL. "Heartbleed" vulnerability has been made available publically; any service using a vulnerable version of OpenSSL is very likely to be susceptible to attack. Another name for Heartbleed is zero –day vulnerability. Mitigating Heartbleed bug which faces computer systems in the global cyberspace is debatable. Considering the long exposure, ease of exploitation and attacks leaving no trace, maintaining security of information in cyberspace is very complex (Heartbleed.com, 2014; Nfuka et al., 2014). Heartbleed bug is the worst vulnerability found (at least in terms of its negative impact) since commercial traffic began to flow on the Internet. It is a vulnerability discovered in the TLS heartbeat mechanism built into certain versions of the popular OpenSSL library (Seggelmann et al., 2012).

The Heartbleed bug is a name given to vulnerability within the OpenSSL cryptographic library (CVE-2014-0160) used to encrypt communications between web applications; email exchanges, instant messaging clients and some SSL based virtual private network (VPN) connections (Accuvant-Labs, 2014). OpenSSL is one of the technologies employed in online application to create an encrypted communication session using a protocol called "Hypertext Transfer Protocol Secure" (HTTPS) between a user and a website (OpenSSL, 2014). The

Heartbleed bug opens doors for the cyber criminals to extract sensitive data directly from the server's memory without leaving any traces (Kumar, 2014). Heartbleed bug could be used by attackers to steal personal data such as usernames and passwords; and doing so is relatively easy. However one of the biggest concerns is that the vulnerability could be used to steal the private keys which are used to encrypt communications in different websites. Attackers could eavesdrop on communications or set up fake websites which impersonate legitimate websites allowing them access to even more data. Thus, there is a danger of public disclosure of passwords and other important data of users from their web applications by the cyber-criminals before the web application/software/device is patched with OpenSSL version free of the Heartbleed bug (Hosenball and Dunham, 2014).

Thus, information security goals (i.e. confidentiality, integrity and availability) are compromised by the Heartbleed attack, as it leaves no logs on servers/clients for tracing. For example, the Heartbleed attack has resulted in exposure of social security numbers of the Canada Revenue Agency and other personal data (Lister, 2014). During Heartbleed attack, primary key materials such as encryption keys (private keys), certificates are leaked. Certificates and keys are used to ensure confidentiality, permitting only authorized recipients to view protected data. If an attacker compromises a certificate or key, confidentiality is no longer assured (Heartbleed.com, 2014). Thus, information flowing in the cyberspace from attacked web applications is not secure. Confidentiality and privacy is compromised by leaking confidential information; this can compromise integrity of information as attackers can update, delete or commit any unauthorized transactions in cyberspace. While Heartbleed attack is a direct threat to confidentiality and indirectly to integrity, there are also potential implications for availability of information to be compromised (Arbor-networks, 2014). Every organization should be prepared to defend against Denial of Services (DoS) or Distributed Denial of Services (DDoS) attacks intended to cause state exhaustion and service unavailability for SSL-enabled servers, load-balancers, reverse proxies and virtual private network (VPN) concentrators. The purpose of such DoS/DDoS attacks would be to force targeted organizations to re-start their services in order to recover from the DoS/DDoS attacks, thus providing the attackers with a greater chance of capturing leaked private keys (Arbor-networks, 2014). All these effects have created many debates of whether a given web application was affected by the bug or not; and the magnitude of the impact of the bug. In ascertaining the security of the web applications against Heartbleed attack, there is a need of conducting evaluation of web applications security against the Heartbleed bug.

The purpose of this study was to evaluate security of web applications from selected higher education institutions (HEI) in Africa against the Heartbleed attack. The study specifically aimed at determining whether a given web application is vulnerable to Heartbleed attack or not, safety of certificates/keys, and the magnitude of the impact of the bug. Thus, this paper presents the evaluation of security of web applications against Heartbleed bug based on Activity theory and Soft systems methodology (SSM).

2. THEORETICAL FOUNDATION AND CONCEPTUAL FRAMEWORK FOR EVALUATION OF SECURITY OF WEB APPLICATIONS AGAINST HEARTBLEED BUG

The Activity theory and Soft systems methodology were adopted to guide the evaluation of security of web applications against Heartbleed bug.

2.1. Activity Theory: Theoretical foundation for Evaluation of Security of Web Application against Heartbleed Bug

This study applied the Activity theory as theoretical foundation for the study. The study also applied system thinking approach, since researchers were looking for the conceptual framework which fits well on the subject under investigation (Alter, 2004; Turpin and Alexander, 2014). Activity Theory states that human work is always social, cooperative, and collective; and takes place within a division of labour (Gonçalvesa et al., 2013). Activity theory is a descriptive framework for studying the contextual aspects of different practices, linking the individual and social dimensions of that practice. Activity Theory uses the whole work activity as the unit of analysis, where the, activity is broken into the analytical components of subject, tool and object,

where the subject is the person being studied, the object is the intended activity, and the tool is the mediating device by which the action is being executed (Figure 1).

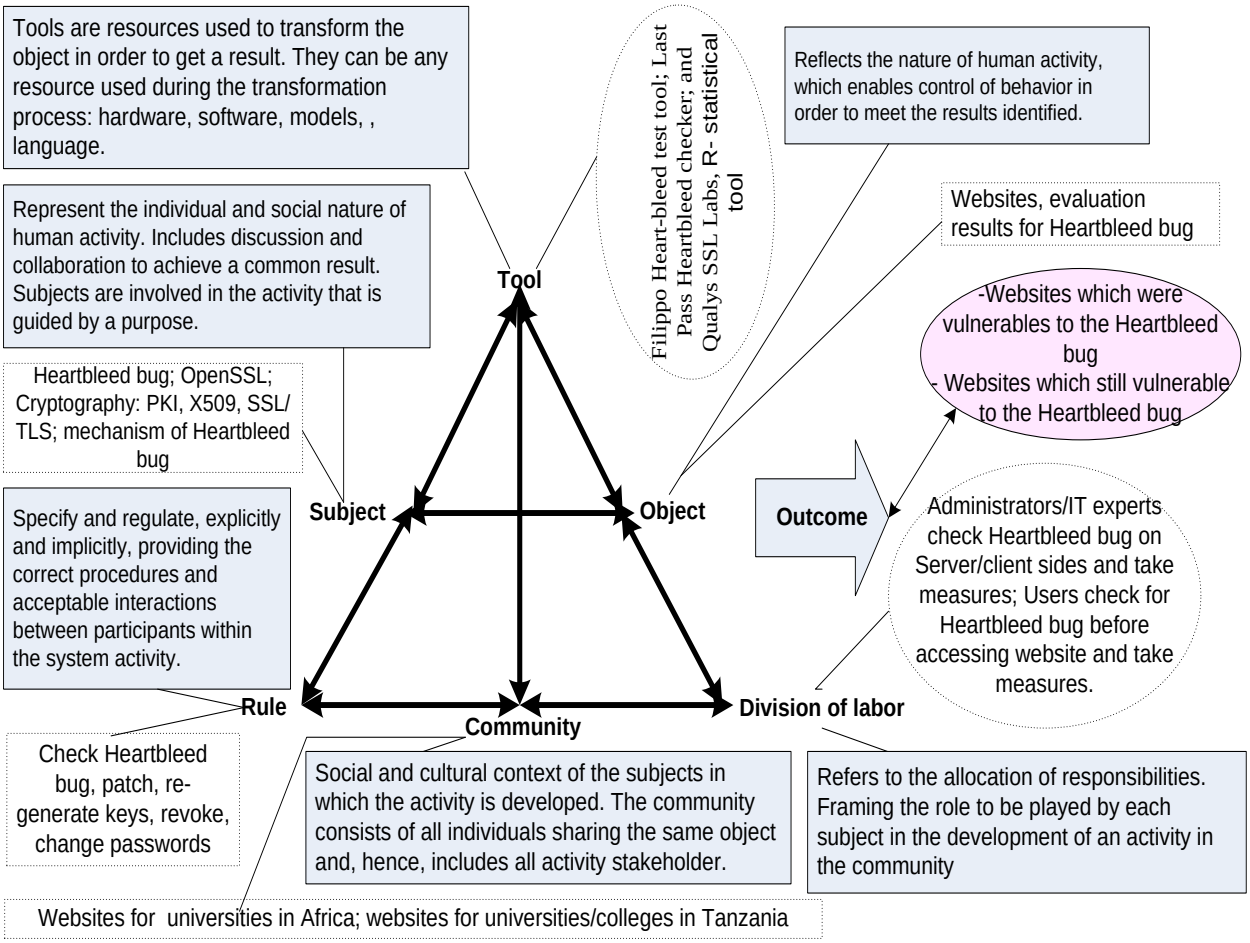


Figure 1: Activity Theory: Theoretical foundation for Evaluation of Security of Web Applications against the Heartbleed Bug (Adapted from Engeström, 2000)

2.2. Soft Systems Methodology (SSM): Conceptual framework for Evaluation of Security of Web Applications against Heartbleed Bug

SSM originally was applied as a modelling tool, but in later years it has been perceived as a learning and development tool. Although it develops models, the models are not supposed to represent the "real world", but by using systems rules and principles allow researcher to structure your thinking about the real world (Checkland, 1981, 1998). At the heart of SSM is a comparison between the world as it is, and some models of the world as it might be. Out of this comparison arise a better understanding of the world ("research"), and some ideas for improvement ("action"). The SSM as contained in the original works of Checkland (1981, 1998) has seven stages. Some of them address the "real world", and some of them addressing a "conceptual world". The seven stages as coined by Checkland (1981) are:

- stage 1: entering the problem situation (problem situation unstructured)

The problem situation is first experienced, as it is, by the researcher. That is, the researcher makes as few presumptions about the nature of the situation as possible.

- stage 2: expressing the problem situation (problem situation structured)

In this step, the researcher develops a detailed description, a "rich picture", of the situation within which the problem occurs.

- stage 3: formulating root definitions of relevant systems (root definition of the relevant system)

In this step, the "root definitions", the essence of the relevant systems, are defined.

- stage 4: building Conceptual Models of Human Activity Systems

This stage consists of the conceptual model, which represents the minimum set of procedures for the system to be able to achieve the desired transformation.

- stage 5: comparing the models with the real world
- Stage 6: defining changes that are desirable and feasible
- Stage 7: action to improve the problem situation

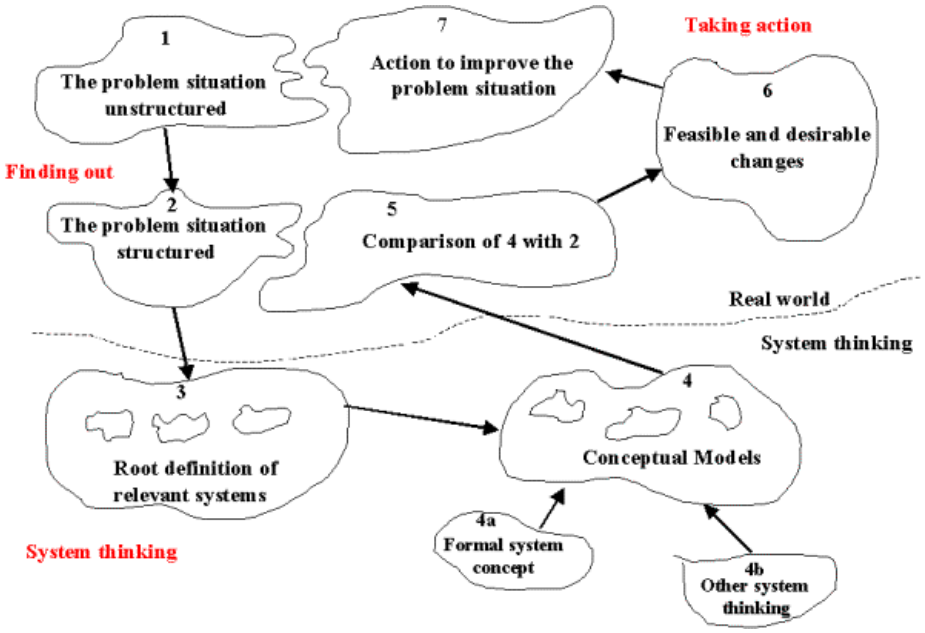


Figure 2: Stages of SSM (Checkland, 1981, 1998)

Applying the seven stages of SSM (Figure 2), Soft systems thinking seeks to explore the 'messy' problematic situations that arise in human activity. The problem situation in this study was "Web applications of HEIs in Africa are vulnerable to Heartbleed attack". The problem situation has been expressed by describing the Heartbleed bug, OpenSSL, underlying technology and infrastructure: Cryptography, Public Key Infrastructure (PKI), X.509 Certificate, SSL/TLS and Mechanisms of Heartbleed bug. The conceptual model (Figure 3) has been developed (using stage 1-3 of SSM). The SSM have been adopted for the management of the analysis of data in a systematic way (Checkland, 1981, 1998; Sanga, 2010; Umoh, 2013).

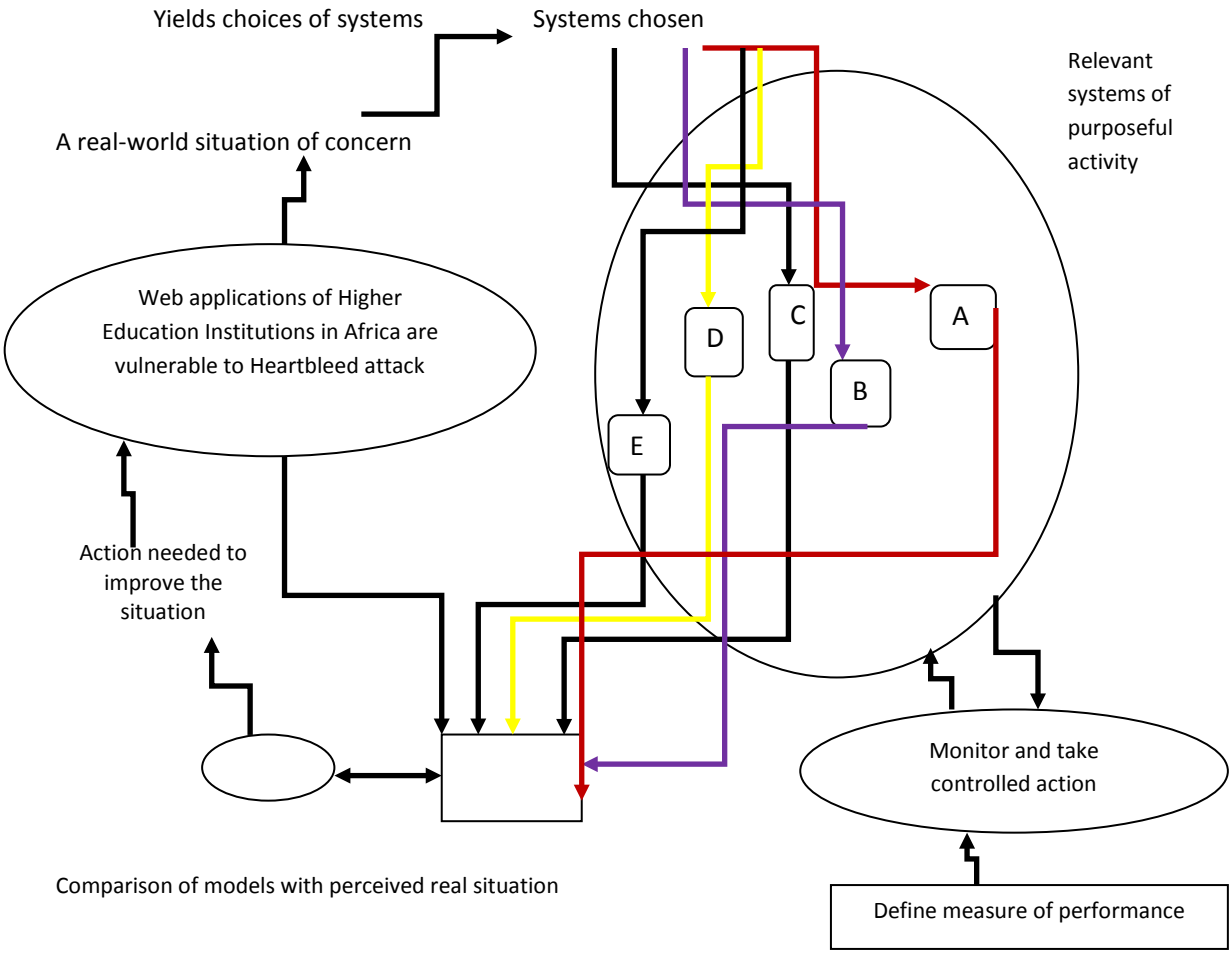


Figure 3: How SSM was used in This Study (Adapted from Checkland and Scholes ,1990; Sanga, 2010)

KEY: 'A' stands for analysis using Filippo Heart-bleed test tool, 'B' stands for analysis using Last Pass Heartbleed checker, 'C' stands for analysis using Qualys SSL Labs, 'D' stands for deep analysis tool for SSL and Heartbleed attack, and 'E' stands for any future study

3. HEARTBLEED BUG

The Heartbleed bug (referred officially as CVE-2014-0160) is a vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet (Heartbleed.com, 2014). OpenSSL is a widely used implementation of the SSL/TLS protocol. Heartbleed may be exploited whether the party using a vulnerable OpenSSL instance for TLS is either a server or a client.

Heartbleed results from improper input validation (due to a missing bounds check) in the implementation of the TLS heartbeat extension (Seggelmann et al., 2012; Cyberoam, 2014; Heartbleed.com, 2014). An overrun vulnerability in the OpenSSL cryptographic library affected around 17% of the Internet's secure SSL web servers which use certificates issued by trusted certificate authorities; allowing theft of the servers' private

keys and users' session cookies and passwords (Netcraft, 2014). It affects any service supported by IMAP, SMTP, HTTP and POP (http://www.kb.cert.org/vuls/id/720951) running over vulnerable OpenSSL (SSL/TLS). This is the reason of being named as the worst vulnerability found since commercial use of the Internet (Steinberg, 2014).

3.1. OpenSSL

The OpenSSL is a commercial-grade, full-featured, and Open Source toolkit implementing the SSL (version 2/ version 3) and TLS (version 1 or later) protocols as well as a full-strength general purpose cryptography library (OpenSSL, 2014). OpenSSL is an open-source implementation of the SSL/TLS protocol and many websites use OpenSSL to achieve security The vulnerable versions of OpenSSL are 1.0.1 through 1.0.1f; while version 1.0.1g, 1.0.0 and 0.9.8 branch are not vulnerable to Heartbleed attack (Heartbleed.com, 2014; OpenSSL, 2014). The Heartbleed bug was introduced to OpenSSL in December 2011 and has been out in the wild since OpenSSL release 1.0.1 on 14th of March 2012. OpenSSL 1.0.1g was released on 7th of April 2014 with fixed bug. However this bug has left large amount of private keys and other secrets exposed to the Internet in those web applications which are not fixed. Considering the long exposure, ease of exploitation and attacks leaving no trace thus this bug should be taken seriously (Heartbleed.com, 2014). The best way in order to avoid attacks by Heartbleed bug is to implement a secure cryptography algorithm in web applications (Fsvadvisors, 2014).

3.2. Cryptography

Cryptography (or cryptology) is the art and science of hiding the meaning of a communication from unintended recipients (Krutz and Vines, 2007). Cryptography also allows senders and receivers to authenticate each other through the use of key pairs (Kessler, 2014). The following are types of algorithms for cryptography:

- Secret Key Cryptography (symmetric encryption): It is a type of algorithm where only one key is used for both encryption and decryption. This type of encryption is also referred to as symmetric encryption. Some of the symmetric encryption algorithms are AES (Rijndael), 3DES, RC2, Blowfish, and RC6 (Elminaam et al., 2009; Kessler, 2014).
- Public Key Cryptography (asymmetric encryption): It is a type of algorithm where two keys are used. This type of encryption is also called asymmetric encryption. One key is the public key and anyone can have access to it. The other key is the private key, and only the owner can access it. The sender encrypts the information using the receiver's public key. The receiver decrypts the message using his/her private key. For non-repudiation, the sender encrypts plain text using a private key, while the receiver uses the sender's public key to decrypt it. Thus, the receiver knows who sent it. Some of the asymmetric algorithms used in public key cryptography are RSA (Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman), Diffie-Hellman (DH), and Digital Signature Algorithm (DSA) (Elminaam et al., 2009; Kessler, 2014).
- Hash Functions: It is a type of algorithm where there is no key at all and also called one-way encryption. Hash functions are mainly used to ensure that a file remain unchanged (Elminaam et al., 2009; Kessler, 2014). The weakness of cryptography algorithms can be addressed by the public key infrastructure.

3.3. Public Key Infrastructure (PKI)

PKI is an integration of digital signatures and certificates, and the other services required for e-commerce (Krutz and Vines, 2007). The PKI is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates. In cryptography, a PKI is an arrangement that binds public keys with respective user identities by means of a certificate authority (CA). The PKI includes digital certificates, certificate authority (CA), registration authorities, policies and procedures, certificate revocation, non-repudiation support, time stamping, Light Directory Access Protocol (LDAP) and security enabled applications (Krutz and Vines, 2007; Netcraft, 2014).

PKI enables users of a basically unsecure public network such as the Internet to secure and exchange data privately and money through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority. The PKI provides a digital certificate that can identify an individual or an organization and directory services that can store and, when necessary, revoke the certificates (Articsoft, 2013). In public key cryptography, a public and private key are created simultaneously using the same

algorithm (a popular one is known as RSA) by a CA. The private key is given only to the requesting party and the public key is made publically available (as part of a digital certificate) in a directory that all parties can access. The private key is never shared with anyone or sent across the Internet. You use the private key to decrypt text that has been encrypted with your public key by someone else (who can find out what your public key is from a public directory) (Searchsecurity.techtarget.com, 2014).

The PKI is mainly used for signing; private key to "sign data"; verification: use public key to verify "signature"; encryption: use public key to encrypt data; decryption: use private key to decrypt to data (Zygadlo, 2009). Internet users rely on public key cryptography to verify the identity of secure websites and SSL certificates contain a public key that is generated from its associated private key. At the start of the secure connection, the server proves that it has the private key by decrypting messages encrypted with the public key, or by cryptographically signing its own messages. Keeping the private key secret is critical, if an attacker steals the private key; the attacker can impersonate the secure website, decrypt sensitive information, or perform a man-in-the-middle attack (Netcraft, 2014). In order to secure the CA in a directory of PKI for Internet based application using a digital signature, there is a need of securing it using X.509 certificate.

3.4. X.509 Certificate

The original X.509 certificate was developed to provide the authentication foundation for the X.500 directory (Krutz and Vines, 2007). X.500, the Open Systems Interconnection model (OSI) directory standard, defines a comprehensive directory service, including an information model, a namespace, a functional model, and an authentication framework (Venkatakrishnan, 2014). X.509 is one of many standards for PKI; determines a format for certificates, keys, revocations, and others pieces. X.509 is a widely used standard for defining digital signatures. Internet Engineering Task Force (IETF) Public Key Infrastructure X.509 (PKIX) has developed a profile based on X.509 for Internet use (Figure 4).

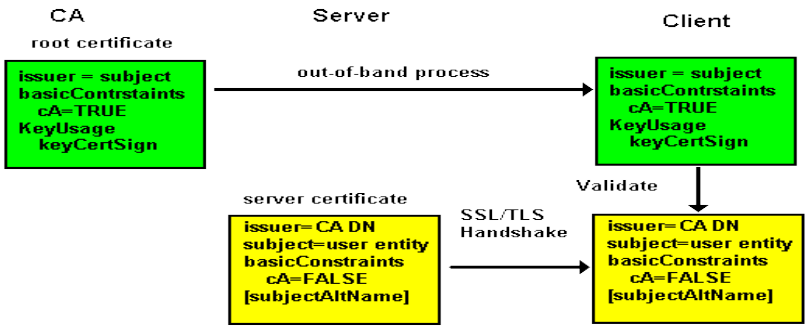


Figure 4: X.509 Usage (Zytrax.com, 2014)

In case PKI is affected by Heartbleed bug it is recommended to replace all keys and certificates, because you don't know which systems even non OpenSSL ones may have had keys and certificates stolen via stepping-stone attacks. Thus, the best option is to assume all keys and certificates have been stolen. The average large enterprise can have more than 17,000 encryption keys and certificates (Hill, 2014). When extrapolating the cost to respond to the Heartbleed vulnerability in such enterprise, it costs the organization $115.00 per certificate. In order to replace 17,000 encryption keys and certificates it will cost an organization $1.95 million in labour costs alone (Hill, 2014). It is advised that do not reuse the same private key; by reusing the same private key, a site that was affected by the Heartbleed bug still faces exactly the same risks as those who have not yet replaced their SSL/TLS certificates; if the previous certificate had been compromised, then the stolen private key can still be used to impersonate the website's new SSL certificate, even if the old certificate has been revoked. Certificates that have been reissued with the same private key are easy to identify, as the new public key will also be identical to the old SSL/TLS certificates (Netcraft, 2014).

3.5. SSL/ TLS

The SSL protocol was developed by Netscape in 1994 to secure Internet client-server transactions (Krutz and Vines, 2007). TLS and its predecessor, SSL, are cryptographic protocols designed to provide communication

security over the Internet (Venkatakrishnan, 2014). Cryptographic protocols are small distributed algorithms that aim to provide some security related objective over a public communication network, such as the Internet (Smyth, 2011). The TLS protocol was released in January 1999 by IETF as a standard for private communications over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering or message forgery. Many websites use SSL/TLS protocol to protect confidential user information such as credit card numbers, social security numbers, and login credentials to be transmitted securely (Luvanda et al., 2014). Many email clients also support SSL/TLS as a means to encrypt messages to keep them safe from identity thieves who watch public wireless networks looking for passwords and credit card numbers. The current version of SSL protocol is SSL 3.0; while for TLS are TLS 1.1 and TLS 1.2. TLS/SSL provides a secure tunnel to a server, which is most commonly authenticated by an X.509 certificate (Clark and Oorschot, 2013). TLS/SSL provides three security measures, namely: client authentication, data encryption and data integrity checks (Dierks and Rescorla, 2008, pp.78-96; Luvanda, Kimani, and Kimwela, 2014).

Client authentication is meant to ensure that the client can uniquely identify the server, and can verify that data transfer will be secure. Data encryption ensures that the data is scrambled using complex encryption algorithms, so that even if it is intercepted in route it cannot be deciphered. Data integrity checks: verifies that there has been no alteration of the data during transit (Dierks and Rescorla, 2008, pp.78-96). TLS/SSL use X.509 certificates, and hence asymmetric cryptography to authenticate the counterparty with whom they are communicating and to exchange a symmetric key. This session key is then used to encrypt data flowing between the parties. This allows for data/message confidentiality and message authentication codes for message integrity and as a by-product, message authentication. TLS/SSL is widely used in applications such as web browsing, electronic mail, Internet faxing, instant messaging, and voice-over-IP (VoIP). The most common protocols which use TLS/SSL are HyperText Transfer Protocol Secured (HTTPS) and Secure File Transfer Protocol (SFTP). TLS/SSL runs on top of TCP but below the end user protocol that it secures such as HTTP, FTP or IMAP (Zytrax.com, 2014) as shown in Figure 5.

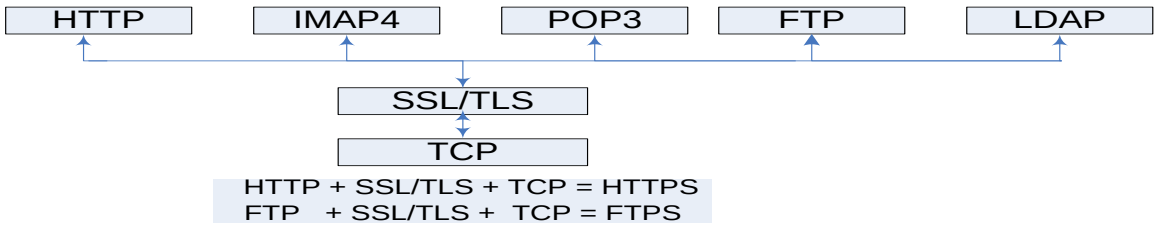


Figure 5: TLS/SSL Layering (Zytrax.com, 2014)

When a secure connection is established using TLS/SSL, for example using HTTPS (default port 443), an exchange of messages occur between the client which always initiates the connection and a server. The first set of messages are called the Handshake Protocol after which both client and server enter the Record (or Data) Protocol (Phillips, 2014; Zytrax.com, 2014) as shown in Figure 6.

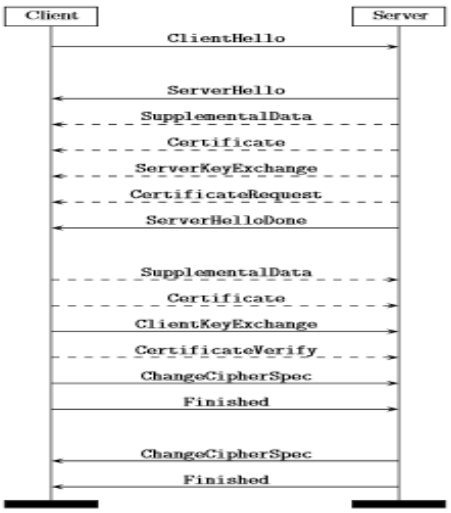


Figure 6: TLS/SSL Protocol Sequences (Zytrax.com, 2014; Phillips, 2014; Patel, 2014)

3.6. Mechanisms of Heartbleed /OpenSSL Bug

Heartbleed bug is the bug that is present in OpenSSL versions 1.0.1 through 1.0.1f. These versions have seen a great deal of adoption lately, because security professionals have been urging developers to implement more recent versions of TLS (1.1 and 1.2) (Heartbleed.com, 2014; OpenSSL, 2014). Deployment of TLS 1.2 currently stands at about 30% of the SSL pulse data set and many of those servers are likely to be vulnerable (Clark and Oorschot, 2013; Cyberoam, 2014). The problem is that there's a simple missing bounds check in the code that handles TLS heartbeat messages which allows an attacker to request that a running TLS server handover a relatively large slice (up to 64KB) of its private memory space. Since this is the same memory space where OpenSSL also stores the server's private key material, an attacker can potentially obtain long-term server private keys, TLS session keys, confidential data like passwords and session ticket keys (Green, 2014). The Heartbleed is the attack which steals OpenSSL private keys, steal OpenSSL secondary keys and retrieve up to 64kb of memory from the affected server. As a result, the attacker can decrypt all traffic between the server and client(s) (Figure 7).

The Heartbleed attack is not a virus; and it is not a flaw in SSL/TLS protocol specification/design but it is an implementation problem; programming mistake in popular OpenSSL library that provides cryptographic services such as SSL/TLS to the applications and services (Heartbleed.com, 2014). The affected versions of hardware/software are those which use vulnerable versions of OpenSSL. Some operating system distributions that have shipped with potentially vulnerable OpenSSL version are Debian Wheezy (stable), OpenSSL 1.0.1e-2+deb7u4; Ubuntu 12.04.4 LTS, OpenSSL 1.0.1-4ubuntu5.11; CentOS 6.5, OpenSSL 1.0.1e-15; Fedora 18, OpenSSL 1.0.1e-4; OpenBSD 5.3 (OpenSSL 1.0.1c 10 May 2012) and 5.4 (OpenSSL 1.0.1c 10 May 2012); FreeBSD 10.0 - OpenSSL 1.0.1e 11 Feb 2013; NetBSD 5.0.2 (OpenSSL 1.0.1e); OpenSUSE 12.2 (OpenSSL 1.0.1c) (NIST, 2014). Affected Cyberoam operating systems versions are 10.6.0 Beta-3, 10.6.1 RC-1, and 10.6.1 RC-3 (Cyberoam, 2014). Many of Cisco routers, modems and networking gear are also affected by Heartbleed bug (Krawczyk, 2014). Others devices/software vulnerable to Heartbleed bug hack attacks are phones and tablets running Android 4.1.1 and any other version which use vulnerable OpenSSL (Goodin, 2014).

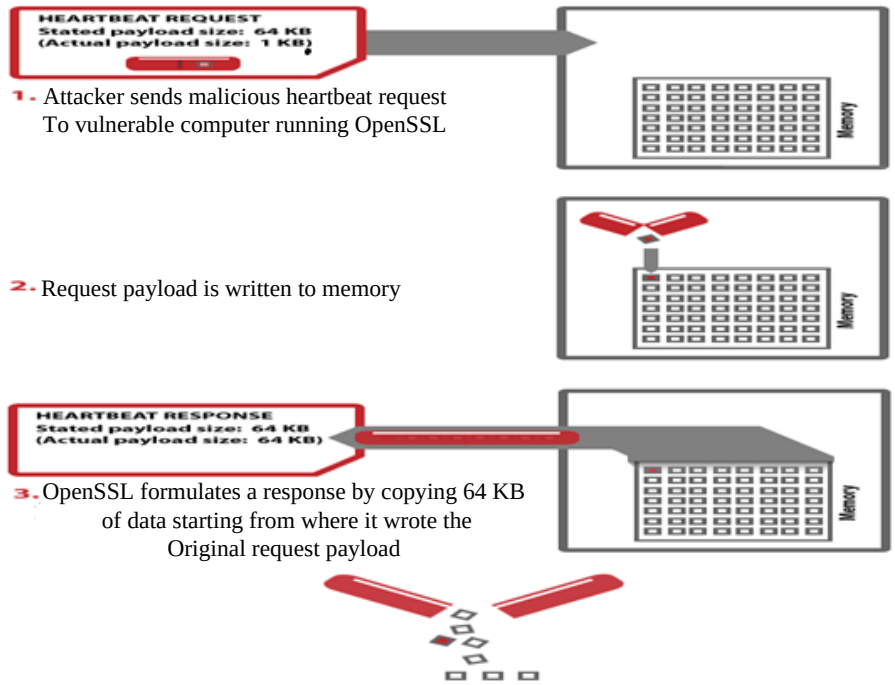


Figure 7: Mechanisms of Heartbleed /OpenSSL Bug (Symantec, 2014)

There have been a number of valuable studies of Heartbleed bug: such as studies by Afidler, Granick and Crenshaw (2014); Chabrow (2014); Fsvadvisors (2014); Mpofu et al., (2014) and others. However, none of these studies were carried out to evaluate the security of web applications against the Heartbleed attack in HEIs from Africa. Thus, there was a need to conduct study for evaluation of the security of web applications against the Heartbleed attack to determine whether web applications from selected HEIs are either affected or vulnerable.

4. MATERIAL AND METHODS

This study employed both quantitative and qualitative research methodologies. The mixed research methodology (triangulation) was chosen, since the weakness of one was complemented by the strength of the other. In this study, case study and content/documentary analysis research methods were used. Robson (2002:178) defines case study as a strategy for doing research which involve empirical investigation of a particular contemporary phenomenon within its real life using multiple sources of evidence. The case study is a worthwhile way of exploring existing theory. In addition, a simple, well-constructed case study was employed to challenge existing theories and also provide a source of new hypotheses (Saunders et al., 2009). Content/documentary analysis of web applications were carried out using various analysis tools for data collections. The tools used were Filippo Heart-bleed test tool; Last Pass Heartbleed checker; and Qualys SSL Labs, deep analysis tool for SSL and Heartbleed attack.

Due to the nature of the research problem, Soft Systems Methodology (SSM) (Figure 3) was adopted for the management of the analysis of data in a systematic way (Sanga, 2010; Umoh, 2013). SSM is a systemic approach for tackling real-world problematic situations in a circular fashion. SSM is the result of the continuing action researches that researchers have conducted over 30 years, to provide a technique for users to deal with the kind of messy problem situations (such as the evaluation of Heartbleed bug) that lack a formal problem definition (Checkland, 1981, 1998). SSM was supported by the Activity Theory as conceptual framework. This framework helped the researchers to use activity as basic unit for analysis. The evaluation of security of web applications against Heartbleed bug was modelled using Activity Theory by expressing in

terms of its six components namely: tool, subject, rule, community, and division of labour, object and outcome (Figure 1).

4.1. Sampling Technique

The sample size for this study comprised of 100 top ranked universities websites in Africa (Webometrics.info, 2014a) and 45 universities/colleges websites from Tanzania (TCU, 2014). Data collection from these sampled websites was done in a circular manner using SSM (Figure 3). The sample size for websites with OpenSSL was 64 (36 websites from among of 100 ranked websites universities in Africa; 28 websites from Tanzania universities/colleges) websites universities in Africa. The universities websites without OpenSSL (SSL/TLS) from Africa were not considered for further analysis in this study. The analysis of the collected data in each cycle was done using "R statistical computing package". R is a software language for carrying out complicated (and simple) statistical analyses. It is a language and environment for statistical computing and graphics (R Core Team, 2013). The choice of R was based on the nature of collected data and capability of R in comparison with other statistical data analysis languages. The findings from this study were described and presented in form of tables, charts, figures and graphs.

5. RESULTS AND DISCUSSIONS

Different organisations are trying to fix their web applications from the Heartbleed bug, after the public announcement of the bug. However fixing the Heartbleed bug by updating to the invulnerable version of OpenSSL does not guarantee elimination of the effect of the Heartbleed bug, unless other measures are taken. Examples of such measures include re-issuing the certificates, and revoking at client and server side plus changing other private information's such as password, credit numbers, etc. The study involved the evaluation of security of 64 web applications of HEIs in Africa which use OpenSSL. Evaluation was carried out, in order to ascertain whether they are free from the Heartbleed attack or not. The results are summarized and discussed in sections 5.1, 5.2, 5.3 and 5.4.

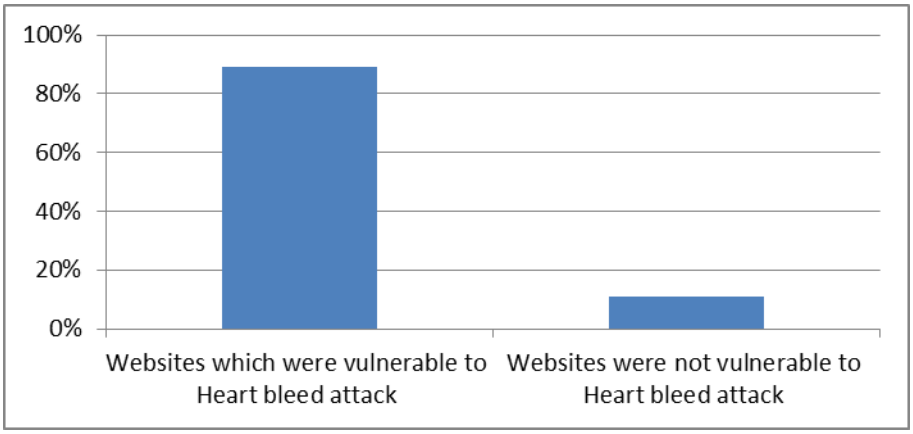
5.1. Evaluation Results for Web Applications Against Heartbleed Attack

The evaluation was carried out to ascertain whether a given web application was vulnerable to Heartbleed bug or not. The results portrayed in Table 1 and Figure 8 depicts that 89% of the most universities web applications were vulnerable to Heartbleed attack; and 11% of the most universities web applications in Africa were not vulnerable to Heartbleed attack. These results shows that most of universities web applications in Africa were vulnerable to Heartbleed attack on public announcement of the bug.

Table 1: Websites which were Vulnerable or Not to Heartbleed Bug

Description	Universities in Africa	
	Quantity	**Percentage**
Websites which were vulnerable to Heart bleed attack	57	89%
Websites were not vulnerable to Heart bleed attack	7	11%
Total	**64**	100%

Source: TCU (2014); Webometrics.info (2014a); Webometrics.info (2014b)



Source: TCU (2014); Webometrics.info (2014a); Webometrics.info (2014b)
Figure 8: Websites which were Vulnerable or Not to Heartbleed Bug

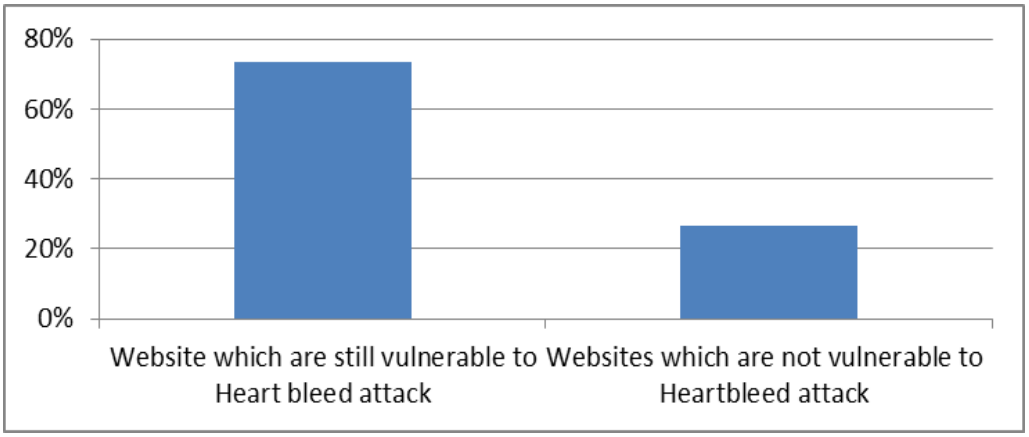
5.2. Websites Which are Still Vulnerable or Not to Heartbleed Attack

The evaluation was carried out to assess the web applications from universities in Africa to ascertain whether are still vulnerable to the Heartbleed attack or not, evaluation was done on June 16, 2014 to July 03, 2014. The result in Table 2 and Figure 9 reveals that 73% of the most universities web applications in Africa are still vulnerable to Heartbleed attack; and 27% of the most universities web applications in Africa are not vulnerable to Heartbleed attack. These results show that many web applications in Africa are still vulnerable to Heartbleed attacks. Different organisations are fixing their web applications against Heartbleed attack, for example one university from Tanzania which was vulnerable on June 16, 2014, but the evaluation of the same website on July 02, 2014 confirmed that it's now safe from the Heartbleed bug.

Table 2: Websites which are Still Vulnerable or Not to Heartbleed Attack

	Universities in Africa	
Description	**Quantity**	**Percentage**
Website which are still vulnerable to Heart bleed attack	47	73%
Websites which are not vulnerable to Heartbleed attack	17	27%
Total	64	100%

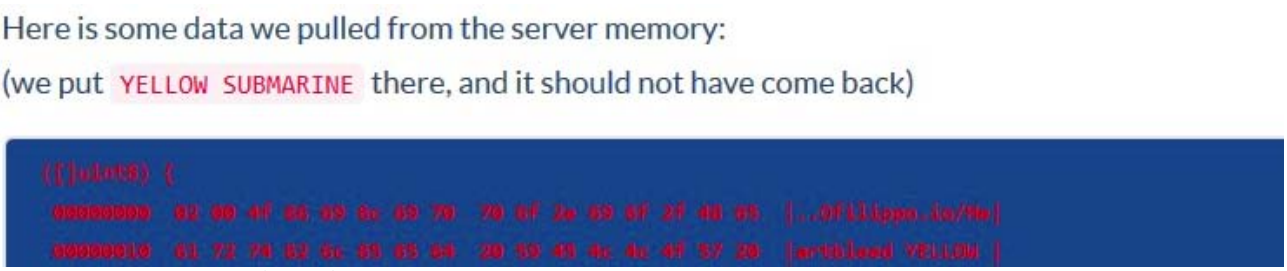
Source: TCU (2014); Webometrics.info (2014a); Webometrics.info (2014b)



Source: TCU (2014); Webometrics.info (2014a); Webometrics.info (2014b)
Figure 9: Websites which are Still Vulnerable or Not to Heartbleed Attack

5.3. Discussions of the Results for Heartbleed Evaluation for Affected Web Applications

The following is the example of evaluation case for a web application which was vulnerable to Heartbleed bug on June 16, 2014 (Figure 10) and the web application was fixed from the bug on June 24, 2014 (Figure 11). The Heartbleed bug evaluation on June 16, 2014 using various tools such as "Filippo Heart-bleed test tool"; "Last Pass Heartbleed checker"; and Qualys SSL LABS deep analysis tool"; revealed that some of the Heartbleed attack vulnerable web applications continually being fixed, even though majority of the affected web applications take longer time to be fixed. For about two months later (Table 1 and Table 2), after the public announcement date of the bug only 16% of the most universities web applications which were vulnerable were patched for the Heartbleed bug. This implies that some organizations have taken actions to fix the Heartbleed bug, while others has taken no measures against the Heartbleed bug. For example, website of "X" University from Tanzania which was found to be vulnerable to the Heartbleed bug on June 16, 2014 (Figure 10); has been fixed from the Heartbleed bug and is now safe, this was revealed by evaluation results on July 02, 2014 (Table 3, Figure 11 to Figure 13).



Source: Lastpass.com (2014a)

Figure 10: Example of Evaluation Case for Vulnerable Website to Heartbleed Bug

Evaluation by "Last Pass Heartbleed checker" tool portrays that the University "X" website is safe from Heartbleed bug from June 24, 2014: 06:33GMT (Table 3).

Table 3: Example of Evaluation Case for Fixed Website from Heartbleed Bug

Server software:	Apache
Was vulnerable:	Possibly (known use OpenSSL, but might be using a safe version)
SSL Certificate:	Now Safe (created 1 week ago at Jun 24 06:33:09 2014 GMT)
Assessment:	Change your password on this site if your last password change was more than 1 week ago

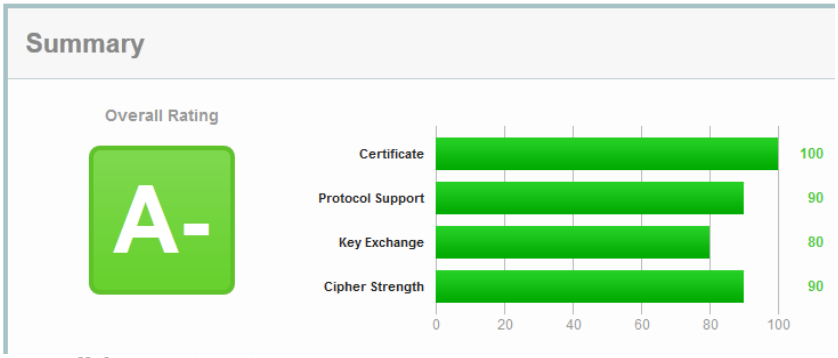
Source: Lastpass.com (2014b)

The Heartbleed bug evaluation on July 02, 2014 using "Filippo Heart-bleed test tool" gave the same results that the University "X" website has been patched from Heartbleed bug (Figure 11).

Source: Filippo.io (2014)

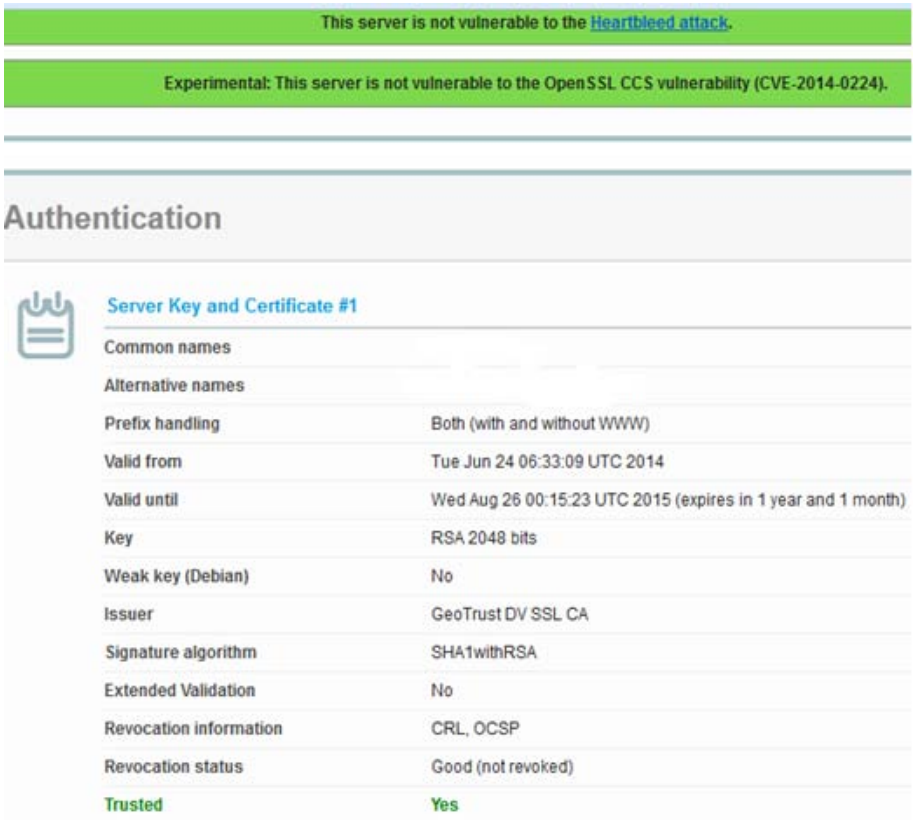
Figure 11: Example of Evaluation Case for Fixed Website from Heartbleed Bug

Thereafter, testing for Heartbleed bug was done for University "X" website using "Last Pass Heartbleed checker"; and Qualys SSL Labs deep analysis tool and the results confirm that it has fixed the website from the Heartbleed bug (Figure 12 and Figure 13).



Source: Sllabs.com (2014)

Figure 12: Summary of Analysis of Fixed Website from Heartbleed Bug



Source: Sllabs.com (2014)

Figure 13: Server Key Analysis for Fixed Website from Heartbleed Bug

5.4. Discussions of the Results for Heartbleed Evaluation and Avoiding/Mitigation Strategies

The results in Table 4 reveals that some HEIs responded immediately after public announcement of Heartbleed bug by taking various security measures to avoid/mitigate the bug, such as patching the systems (servers, clients, network/security appliances), re-generating/re-issuing SSL Certificates, revoking the old SSL Certificates, changing other privacy information such as passwords, credit numbers, etc. Other security measures for avoiding/ mitigating the Heartbleed attack is to conduct awareness to organization regarding the bug; and inform the customers regarding the security measures against the bug. The awareness should involve the importance of testing a given web application for Heartbleed bug before providing any privacy information such as username, password, credit card numbers, social security numbers, etc.

Table 4: Sample of Detailed Analysis for Web Applications of Universities in Africa

S/N	INSTITUTION	SITE	SERVER SOFTWARE	WAS VULNERABLE	IS STILL VULNERABLE	SSL CERTIFICATE	ASSESSMENT
1	A	HEI from Nigeria	Apache/2.2.25 (Unix) mod_ssl/2.2.25 OpenSSL/1.0.0-fips mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635	NOT	NOT	Safe (regenerated 2 years ago)	This server was not vulnerable, no need to change your password unless you have used it on any other site!
2	B	HEI from Tanzania	Apache/2.2.26 (Unix) mod_ssl/2.2.26 OpenSSL/1.0.1e-fips DAV/2 mod_bwlimited/1.4	YES	NOT	Now Safe (created 1 month ago at May 27 21:24:46 2014 GMT)	Change your password on this site if your last password change was more than 1 month ago
3	C	HEI from Nigeria	Apache/2.2.27 (Unix) mod_ssl/2.2.27 OpenSSL/1.0.1e-fips mod_bwlimited/1.4	YES	NOT	Now Safe (created 2 months ago at Apr 7 2014)	Change your password on this site if your last password change was more than 2 months ago
4	D	HEI from Tanzania	Apache/2.2.22 (Unix) mod_ssl/2.2.22 OpenSSL/1.0.0-fips DAV/2 mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635	NOT	NOT	Safe (regenerated 11 months ago)	This server was not vulnerable, no need to change your password unless you have used it on any other site!
5	E	HEI from Tanzania	Apache Phusion_Passenger/4.0.10 mod_bwlimited/1.4 mod_fcgid/2.3.9	YES	NOT	Now Safe (created 2 months ago at May 1 04:34:27 2014 GMT)	Change your password on this site if your last password change was more than 2 months ago
6	F	HEI from Zimbabwe	Apache/2.2.15 (CentOS)	YES	YES	Possibly Unsafe (created 1 year ago at Jan 7 08 2013	Patch the system, re-generate the certificates/keys and change your password
7	G	HEI from Tanzania	Apache/2.2.27 (Unix) mod_ssl/2.2.27 OpenSSL/1.0.1e-fips DAV/2 mod_bwlimited/1.4 mod_qos/10.10	YES	YES	Possibly Unsafe (created 3 months ago at Mar 28 01:34:27	Patch the system, re-generate the certificates/keys and change your password

S/N	INSTITUTION	SITE	SERVER SOFTWARE	WAS VULNERABLE	IS STILL VULNERABLE	SSL CERTIFICATE	ASSESSMENT
						2014 GMT)	

Source: TCU (2014); Webometrics.info (2014a); Webometrics.info (2014b)

The results discussed in sections 5.1, 5.2, 5.3 and 5.4 are in line with other evaluation results conducted by other researchers; for example at AVG's Virus Labs; they found that 12,043 (1.5 per cent) of 800,000 sites in the world were still vulnerable on May 20, 2014; also some government run websites in Asia and Brazil were still at risk (Leyden, 2014; Lioupras and Manthou, 2014; Phillips, 2014). Thus, security measure need to be taken to address the Heartbleed attack; such measures include: patching with software free from the Heartbleed Bug and taking other security measure as recommended in this study.

6. CONCLUSIONS

This study found that 89% of the universities web applications in Africa were vulnerable to the Heartbleed attack; and 11% of the universities web applications in Africa were not vulnerable to Heartbleed on the public announcement of the bug. 16% of the universities web applications which were vulnerable were patched for the Heartbleed bug after two months since the public announcement of the bug. As security prevention strategy, in order to maintain the safety of HEIs website, there is a need to take appropriate security measures on each website component using a combination of systems thinking approach (i.e. SSM) and activity theory. This study has contributed towards the application of systems thinking approach (Alter, 2004; Turpin and Alexander, 2014) in analysing security of web applications. Web applications, however, tend to be uniquely customized; you need to secure each web application according to its Internet service, this can be done by hardening TLS/SSL (Dreyfus, 2014).

7. RECOMMENDATIONS

The following are recommendation regarding security measure for mitigating/avoiding the Heartbleed attack based on the findings from this study:

i. System administrator must patch the system(s)/service(s) that use OpenSSL with the version which is free from the Bug; such as OpenSSL 1.0.1g; or later version of OpenSSL.
ii. System administrator must regenerate and replace all keys; and certificates; because you don't know which systems even non OpenSSL ones may have had keys and certificates stolen via stepping-stone attacks. This is done in assumption that all keys and certificates have been stolen. Old keys should be revoked and do not reuse the same private key.
iii. End users of any web applications must test the website before visiting for the Heartbleed vulnerability; by using Heartbleed test tools recommended in this study.
iv. End users of any web applications must change the password(s) for web based application in cyberspace. The use untraceable password is highly recommended (http://www.rohnfinancial.com/files/45863/RFS-Simple%20Guide-Heartbleed.pdf).
v. System administrator must deploy TLS/ Datagram Transport Layer Security (DTLS) honeypots that entrap attackers and alert about exploitation attempts.
vi. System administrator must either develop or deploy or install appropriate Intrusion Detection (IDS)/ Intrusion Prevention System (IPS) that can detect and prevent the Heartbleed attack.
vii. System administrator must develop rules and standards for software inspection and software quality assurances; from development up to deployment stage.
viii. Government, organizations and universities should enact cyber vulnerability policy to guide in event of cyber-attacks from web applications.

ix. Tanzania should establish National cyber-Security Agency to deal with coordination in case of cyber-attack. Other stakeholders to work with National cyber-Security Agency are the Ministry of Defence, research institutes, HEIs, Ministry of Home affairs and other line ministries (e.g. Ministry of Foreign Affairs and International Co-operation).

8. REFERENCES

ACCUVANT-LABS, 2014. Heartbleed Bug Advisory (CVE-2014-0160). [Online] Accuvant-Labs Available at: http://accuvantstorage.blob.core.windows.net/web/file/2016b4dc040c49ee991b5721e0dd62b3/HeartBleed-Bug-CVE-2014-0160-release.pdf [Accessed June 2014].

AFIDLER, M., GRANICK, J. AND CRENSHAW, M., 2014. Anarchy or Regulation: Controlling The Global Trade in Zero-Day Vulnerabilities. Master Thesis. Stanford University, URL: https://d1x4j6omi7lpzs.cloudfront.net/live/wp-content/uploads/2014/06/Fidler-Zero-Day-Vulnerability-Thesis.pdf.

ARBOR-NETWORKS, 2014. The Heartburn Over Heartbleed: OpenSSL Memory Leak Burns Slowly. [Online] Available at: http://www.arbornetworks.com/asert/2014/04/heartbleed/ [Accessed 22 July 2014].

ARTICSOFT, 2013. An Introduction to PKI (Public Key Infrastructure). [Online] Available at: http://www.articsoft.com/public_key_infrastructure.htm [Accessed 26 June 2014].

CHECKLAND, P., 1981, 1998. Systems Thinking, Systems Practice. Chichester: John Wiley and Sons.

CLARK, J. AND OORSCHOT, P.C.V., 2013. SOK: SSL and HTTPS:Revisiting past challenges and evaluating certificate trust model enhancements. IEEE Symposium on Security and Privacy, pp.511-25.

CVE, 2014. Common Vulnerabilities and Exposures (CVE): The Standard for Information Security Vulnerability Names. [Online] Available at: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=web [Accessed 05 August 2014].

CYBEROAM, 2014. Security Advisory –Heartbleed Vulnerability in OpenSSL. Cyberoam.

DIERKS, T. AND RESCORLA, E., 2008. RFC5246:The Transport Layer Security (TLS) Protocol Version 1.2. Protocol. Network Working Group;URL:http://tools.ietf.org/html/rfc5246.

DIPERT, B., 2014. Heartbleed: the Wakeup Call the Open-Source Community Needed? EDN Network.

DREYFUS, E., 2014. TLS Hardening. Arxiv Preprint Arxiv:1407.2168. Originally Published in BSD Magazine , June 2014 Issue –http://bsdmag.org, URL:http://arxiv.org/pdf/1407.2168.pdf.

ELMINAAM, D.S.A., KADER, H.M.A. AND HADHOUD, M.M., 2009. Performance evaluation of symmetric encryption algorithms. IBIMA Publishing, Vol.8, pp.58-64.

ENGESTRÖM, Y., 2000. Activity theory as a framework for analyzing and redesigning work. Ergonomics, Vol.43(Issue 7), pp.960-974.

FILIPPO.IO, 2014. Heartbleed test. [Online] Available at: https://filippo.io/Heartbleed/#udsm.ac.tz [Accessed 02 July 2014].

FSVADVISORS, B.K., 2014. Heartbleed: Serious Security Vulnerability Serious Wake-up Call. Technical report. Electronic Engineering Journal.

GONÇALVESA, A., SOUSA, P. AND ZACARIAS, M., 2013. Using DEMO and activity theory to manage organization change. Procedia Technology, Vol.1, pp.563 – 572.

GONÇALVES, A., SOUSA, P. AND ZACARIAS, M., 2013. Using Ψ theory and activity theory to management organization change. International Journal of Research in Business and Technology, Vol.3(Issue 1), pp.89-97.

GOODIN, D., 2014. Vicious Heartbleed Bug Bites Millions of Android Phones, Other Devices. [Online] Available at: http://arstechnica.com/security/2014/04/vicious-heartbleed-bug-bites-millions-of-android-phones-other-devices/ [Accessed 21 July 2014].

GREEN, M., 2014. Attack of the Week: Openssl Heartbleed. [Online] Available at: http://blog.cryptographyengineering.com/2014/04/attack-of-week-openssl-heartbleed.html [Accessed 27 June 2014].

HEARTBLEED.COM, 2014. The Heartbleed Bug. [Online] Available at: http://heartbleed.com/ [Accessed 15 June 2014].

HILL, G., 2014. Have You Budgeted for the Next Heartbleed? [Online] Available at: http://www.venafi.com/blog/post/have-you-budgeted-for-the-next-heartbleed [Accessed 15 July 2014].

HOSENBALL, M. AND DUNHAM, W., 2014. White House, spy agencies deny NSA exploited 'Heartbleed' bug. [Online] Available at: http://www.reuters.com/article/2014/04/11/us-cybersecurity-internet-bug-nsa-idUSBREA3A1XD20140411 [Accessed 24 June 2014].

IPA, 2011. How to Secure Your Website :Approaches to Improve Web Application and Website Security. 5th ed. IT Security Center (ISEC): Information-Technology Promotion Agency (IPA).

IPA, 2013. Reporting Status of Vulnerability-Related Information About Software Products and Websites:4th Quarter of 2013. Technical Report. Information-Technology Promotion Agency, Japan (IPA/ISEC).

KESSLER, G.C., 2014. An Overview of Cryptography. [Online] Available at: http://www.garykessler.net/library/crypto.html [Accessed 13 July 2014].

KRAWCZYK, K., 2014. Which CISCO Routers, Modems And Networking Gear Are Affected By And Safe From The Heartbleed Bug? [Online] Available at: http://www.digitaltrends.com/computing/which-cisco-routers-modems-networking-gear-safe-from-heartbleed-open-bug/#!bjRjZF [Accessed 22 July 2014].

KRUTZ, R.L. AND VINES, R.D., 2007. The CISSP and CAP Prep Guide. Platinum Edition. New Delhi: Wiey Publishing Inc.

KUMAR, M., 2014. Heartbleed: Most Frequently Asked Questions. [Online] Available at: http://thehackernews.com/2014/04/heartbleed-bug-explained-10-most.html [Accessed 22 June 2014].

KUMAR, S. AND GADE, R.S.R., 2011. Experimental evaluation of juniper network's netscreen-5GT security device against layer4 flood attacks. Journal of Information Security, Vol.2, pp.50-58.

LASTPASS.COM, 2014a. Lastpass Heartbleed Checker. [Online] Available at: https://lastpass.com/heartbleed/?h=udsm.ac.tz [Accessed 16 June 2014].

LASTPASS.COM, 2014b. Lastpass Heartbleed Checker. [Online] Available at: https://lastpass.com/heartbleed/?h=udsm.ac.tz [Accessed 02 July 2014].

LEYDEN, J., 2014. AVG on Heartbleed: It's Dangerous to Go Alone.. [Online] Available at: http://www.theregister.co.uk/2014/05/20/heartbleed_still_prevalent [Accessed 09 July 2014].

LISTER, J., 2014. US Spy Policy May Put Public PCs At Risk. [Online] Available at: http://www.infopackets.com/news/9006/us-spy-policy-may-put-public-pcs-risk [Accessed 24 June 2014].

LUVANDA, A., KIMANI, S. AND KIMWELA, M., 2014. Identifying threats associated with man-in-the-middle attacks during communication between a mobile device and the back end server in mobile banking applications. IOSR Journal of Computer Engineering (IOSR-JCE), Vol.10(Issue 2), pp.35-42.

MPOFU, T.P., ELISA, N. AND GATI, N., 2014. The heartbleed bug: an open secure sockets. International Journal of Science and Research (IJSR), Vol.3(Issue 6), pp.1470-73. Available at: http://www.ijsr.net/archive/v3i6/MDIwMTQ0ODk=.pdf [Accessed 22 July 2014].

NETCRAFT, 2014. Keys Left Unchanged In Many Heartbleed Replacement Certificates. [Online] Available at: http://news.netcraft.com/archives/2014/05/09/keys-left-unchanged-in-many-heartbleed-replacement-certificates.html [Accessed 15 July 2014].

NFUKA, E.N., SANGA, C. AND MSHANGI, M., 2014. Cybercrimes affecting information systems in the global: is this a myth or reality in tanzania? International Journal of Information Security Science, Vol.3(Issue 2), pp.182-199. Available at: http://ijiss.org/ijiss/index.php/ijiss/article/view/72 [Accessed 22 July 2014].

NIST, 2014. Vulnerability Summary for CVE-2014-0160. [Online] Available at: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0160 [Accessed 21 July 2014].

OPENSSL, 2014. Welcome to the OpenSSL Project. [Online] Available at: https://www.openssl.org/ [Accessed 26 June 2014].

PATEL AND AKASH, 2014. Incorporating privacy and security features in an open source search engine a Project Report Presented to (2014). Master's Projects. Paper 362. http://scholarworks.sjsu.edu/etd_projects/362.

PHILLIPS, M., 2014. TLS Filter: An Application-Level Firewall for Transport Layer Security. Technical Report. URL:http://www.doc.ic.ac.uk/teaching/distinguished-projects/2014/m.phillips.pdf.

SANGA, C., 2010. A Technique for the Evaluation of Free and Open Source E-Learning Systems. PhD Thesis. University of the Western Cape.

SAUNDERS, M., LEWIS, P. & THORNHILL, A., 2009. Research Methods for Business Students. 5th ed. London: Pearson Education Limited.

SEARCHSECURITY.TECHTARGET.COM, 2014. PKI (public key infrastructure). [Online] Available at: http://searchsecurity.techtarget.com/definition/PKI [Accessed 13 July 2014].

SEGGELMANN, R., TUEXEN, M. AND WILLIAMS, M.G., 2012. ISSN: 2070-1721 RFC 6520: Transport layer security (tls) and datagram transport layer security (dtls) heartbeat extension. RFC. Internet Engineering Task Force (IETF). Category: Standards Track.

SHEMA, M., 2011. Web Application Security for Dummies. The Atrium, Southern Gate, Chichester, West Sussex, England: John Wiley AND Sons, Ltd.

SMYTH, B., 2011. Formal Verification of Cryptographic Protocols With Automated Reasoning. PhD Thesis. School of Computer Science, College of Engineering and Physical Sciences, University of Birmingham.

STEINBERG, J., 2014. Massive Internet Security Vulnerability. [Online] Available at: http://www.forbes.com/sites/josephsteinberg/2014/04/10/massive-internet-security-vulnerability-you-are-at-risk-what-you-need-to-do/ [Accessed 26 June 2014].

SYMANTEC, 2014. Heartbleed Bug Poses Serious Threat to Unpatched Servers. [Online] Available at: http://www.symantec.com/connect/blogs/heartbleed-bug-poses-serious-threat-unpatched-servers [Accessed 01 July 2014].

TAIPEITIMES, 2014. Millions of Android devices vulnerable to Heartbleed. [Online] Available at: http://www.taipeitimes.com/News/biz/archives/2014/04/14/2003587959 [Accessed 21 July 2014].

TCU, 2014. Registered Institutions: List of Registered Local Fully Fledged Universities, Constituent College and Centres. [Online] Available at: http://www.tcu.go.tz/images/pdf/Recognised_Universities_Colleges_Centres.pdf [Accessed 13 July 2014].

UMOH, G.I., 2013. The use of soft systems methodology (ssm) in evaluating the tourism industry in Nigeria: prospects and challenges. International Journal of Business and Business Management Review, Vol.1(Issue 3), pp. 111-127.

VENKATAKRISHNAN, R., 2014. Redundancy-Based Detection of Security Anomalies in Web-Server Environments. Master Thesis. North Carolina State University, URL:http://repository.lib.ncsu.edu/ir/bitstream/1840.16/9468/1/etd.pdf.

WEBOMETRICS.INFO, 2014a. Ranking Web of Universities:Africa. [Online] Available at: http://www.webometrics.info/en/Africa?sort=asc&order=Excellence%20Rank* [Accessed 13 July 2014].

WEBOMETRICS.INFO, 2014b. Ranking Web of Universities: Tanzania. [Online] Available at: http://www.webometrics.info/en/Africa/Tanzania%2C%20United%20Republic%20of?page=0 [Accessed 13 July 2014].

ZYGADLO, Z., 2009. Interoperability for Electronic ID. Master Thesis. IBIMA Publishing.

ZYTRAX.COM, 2014. Survival Guides - TLS/SSL and SSL (X.509) Certificates. [Online] Available at: http://www.zytrax.com/tech/survival/ssl.html [Accessed 06 July 2014].