International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 3, No.3 (Sep-2014)

An Approach for Systematically Analyzing and Specifying Security Requirements for the Converged Web-Mobile Applications

Devotha Nyambo[1], Zaipuna Yonah[2] and Charles Tarimo[3]

[1, 2] Nelson Mandela African Institute of Science and Technology, Arusha, Tanzania
[3] Univeristy of Dar es Salaam, Dar es Salaam, Tanzania

Received 4 Apr. 2014, Revised 29 Apr. 2014, Accepted 31 May. 2014, Published 1 Sep. 2014

Abstract: As the use of web and mobile applications is becoming pervasive for service delivery and user mobility support, enterprises are now increasingly fighting against a huge number of emerging security threats which interfere with the process of service delivery. As an attempt to help the enterprises in dealing with the emerging security threats in the converged service delivery architecture, this paper presents a methodology for security threat analysis and security requirements specification in web/mobile applications development. The presented methodology is based on a case study Livestock Data Center (LDC) system, which is being developed and it allows both web and mobile interfaces as service delivery channels. Hence the system serves as a representative of other similar setups of service delivery.

In addition to the processes of analysis and security specification, the methodology involves threat modeling as well. There are several threat models in the literature. The STRIDE threats model is one among the existing threats models that is used to identify security threats that needs to be addressed in systems such as the LDC system. The STRIDE threats model has been used to identify the likely security threats to our case study. On applying the STRIDE threats model the following threats were identified as prominent: sensitive data exposure, weak server side controls, client side injection, and weak authentication and authorization.

The identified security threats were compared to existing threats in traditional web and mobile applications separately in order to figure out the changes when the two computing platforms come together. The findings from our case study have shown that the proposed methodology for security threat analysis and security design can be useful in security requirements specifications in the converged web-mobile applications during development, and can be generally used to assist developers of other similar systems.

Keywords: web and mobile applications security, STRIDE, Livestock Data Center, security requirements.

1. INTRODUCTION

The convergence of web and mobile applications has created a number of security concerns due to the fact that user mobility has become increasingly supported by smart phones and Personal Digital Assistants (PADs) [1]. New security challenges imposed by these technological advancement can be identified through a number of existing alternatives basing on different parameters. Application security challenges can be identified by either focusing on resources and goals as assets of an organization [2], use of graphical approach such as Unified Modeling Language [3], or use of models such as STRIDE [4]. For the purposes of this paper STRIDE threats model has been selected to identify possible security threats for the anticipated Livestock Data Center system. STRIDE threats model is used due to its potential of identifying security threats with a focus on attacker goals.

The STRIDE threats model provides a threats identification framework with six parameters: Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service and Elevation of privilege. The model works by classifying attacker goals rather than system resources and assets. The experience in working with the tool has shown that, STRIDE model can allow an analyst to look ahead of what can be the goals of an attack to a system [5].

Identification of security threats paused by an application needs to have three major steps: application decomposition, determination and ranking of threats, and determination of countermeasure and mitigation [4]. The third step will not be discussed further in this paper, but

E-mail: nyambod@nm-aist.ac.tz, zaipuna.yonah@nm-aist.ac.tz, charles@udsm.ac.tz

as recommendation for future research. Therefore, major goals of this paper are to produce possible security threats rising from the convergence of web and mobile applications, and to document key security requirements for the LDC system, which also might be applicable in other similar systems.

The rest of this paper presents a description of the Livestock Data Center system, its decomposition (an integration of both web application and mobile application features), and determination and ranking of threats. Lastly, the specification of security requirements for the LDC system is presented.

To this end, the paper is organized into further sections as follows; Section II: identifications of security threats using STRIDE threats model, Section III: specification of the livestock data center system security requirements, Section IV: conclusion and recommendation for future research.

2. IDENTIFICATION OF SECURITY THREATS USING STRIDE THREATS MODEL

A. Selection of STRIDE Threats Model

There are a number of methodologies for identifying security threats in information systems. With relevance to the LDC system, three among the existing alternative methodologies have been considered, discussed and only one selected based on applicability, efficiency and reliability.

The first considered approach for security threats identification is proposed in [3]. The work contributed to threats identification by developing a graphical approach to support the identification, communication and documentation of security threats and risks. The approach through Unified Modeling Language (UML) allow system users, developers and decision makers to practically engage on threats and risks identification regardless of the fact that they have different backgrounds. However, efficiency and reliability of this methodology is questionable since, information systems can be expanded from time to time and threats identification should not depend on previous status of the system.

The second considered approach is that presented in [5] which describes threats modeling by using STRIDE threats model for threats identification under six parameters; Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service and Elevation of privilege. The model works by classifying attacker goals rather than system resources and assets. A descriptive study in [6], concludes that STRIDE threats model is effective and reliable for different types of system models although it is time consuming.

The final considered approach is the security threats identification as done in [2]. This approach is focused on resources and goals as assets of an organization. The framework developed is an extension of SI* framework by including a reasoning technique based on Answer Set programming (ASP). The extended framework is only for threats that are caused by inappropriate permissions assignments. By using an eHealth scenario, the framework was illustrated and revealed a number of threats relating to confidentiality, integrity and availability. The advantage of this framework is that, it does not rely on the level of expertise of the security analyst to detect threats. However, the reliability of the framework to different types of system models is unknown. Identified features differentiating the methodologies are summarized in Table I.

The selection of STRIDE threats model was motivated by the fact that the model works by classifying attacker goals rather than system resources and assets. Through this approach, a security analyst or designer can predetermine the motive of an attacker to the system and hence strengthen the most vulnerable points. Moreover, the STRIDE threats model can be used for specific components of a system, which allows for secure system expansion.

B. The LDC Application Decomposition

The Livestock Data Center system has three primary components; User interface, database and data analysis engine as shown in Fig. 1 shows. The system is designed to have separate interfaces for each category of users, livestock keepers, extension officers, livestock researchers and veterinary doctors. Livestock keepers will be presented with simple and supportive interfaces for daily data uploading, viewing periodical reports and consulting veterinary doctors and extension officers. Other categories of users will be able to view various reports and analyzed data in form of graphs or charts. Furthermore, as a decision support tool, additional functionalities (such as interpretation of data and graphs generated by the system) for livestock researchers, veterinary doctors and extension officers will be included in their interfaces for them to perform their work in a more efficient manner.

TABLE I. SECURITY THREATS IDENTIFICATION METHODOLOGIES COMPARISON

Methodology	Applicability	Efficiency	Reliability
UML	√	×	×
STRIDE	√	√	√
SI*Framework based on ASP	√	√	×

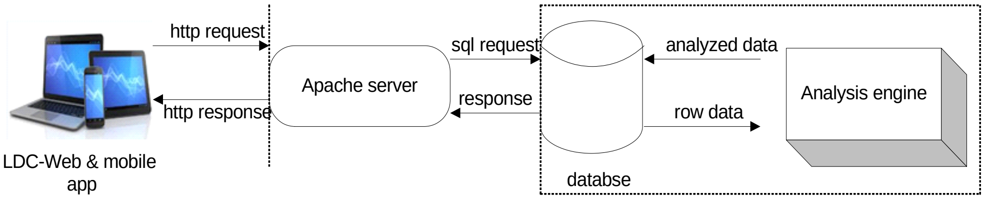

Figure 1. LDC system architectural layout

All data submitted to the system from mobile and web clients will be stored in a database for analysis. This database will be implemented under relational data model and SQL as a back end functionality. Access to the database will be provided to all users depending on specific functionalities such as uploading data and accessing analyzed data for action. Mobile application clients will be able to temporally store data on local SQLite database in case of poor internet connectivity.

The analysis engine is to be designed for the purposes of assisting users to build knowledge from generated reports. The differences will lie on user privileges for accessing reports.

Decomposition of the case study (LDC) is aimed at providing a clear picture of how the application is intended to work and interact with users, and identification of assets that attackers might be interested in. A number of approaches exist for decomposing an application including, use case diagrams, Unified Modeling Language class diagrams, Entity relationship diagrams and Data Flow Diagrams (DFD). Depending on the need for application decomposition, an approach may be selected from among the mentioned and other existing alternatives.

For the case of LDC system security threats identification, we have considered a decomposition that will clearly show all system's processes, data flow, entities, and data stores. Application decomposition by using Data Flow Diagrams (DFD) allows an analyst to identify all system assets, entry points, and entities before modeling the threats [7]. Symbols and notations used in the Data Flow Diagrams are indicated in Table II.

The decomposition has been done in two levels: DFD level 0 and DFD level 1. The Data Flow Diagrams have been drawn only for key features of the LDC system. For this reason, features sharing common assets and users have not been included.

TABLE II. DATA FLOW DIAGRAM SYMBOLS

Symbol	**Description**
▭	An entity/interactor
⊟	process
⊏	Data store
→	Data flow
– – – – – – – – –	Trusted boundary

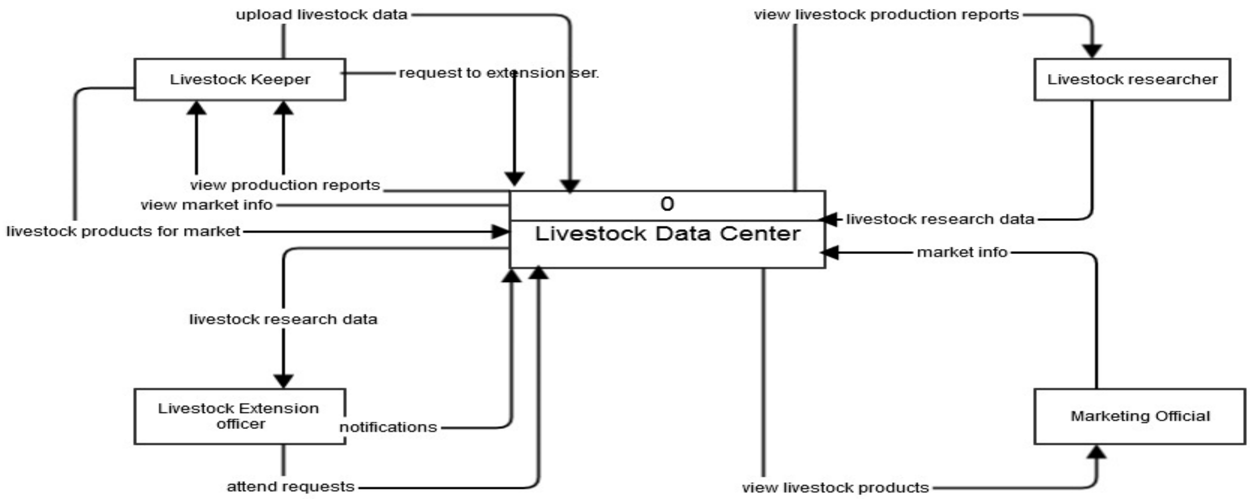

Figure 2. The context level (DFD level 0)

The context diagram, Fig. 2, shows that users of the LDC system are categorized into four groups: Livestock keeper, livestock extension officer, livestock researcher and veterinary doctor. The diagram depicts major interactions among users and the system, however, specific system assets need to be described at DFD level 1 shown in Fig. 3.

Identified system resources and assets include data sources, processes, data flow and user interactions. These resources and assets are briefly described in the following paragraphs:

- *Data sources* - Identified data sources of the LDC system includes: the system users (livestock keepers, livestock researchers, livestock extension officers and marketing officials) and system databases. In the context of web/mobile applications, protecting data sources is becoming a critical issue in the current state of art. Application developers are less involved into data protection while the risks imposed by external sources like application owners, application stores, Operating System and device manufacturers, and third party applications like advertising providers is of great significance [8], [9], [10]. Collaboration in the applications ecosystem is of great concerns to data security because, there is a chain of a number of actors that demands a significant level of security controls to ensure data and data sources protection. Developers in the context of mobile applications can cause a significant risk to users or loss of data integrity for data stored on user's device or remote servers [10].

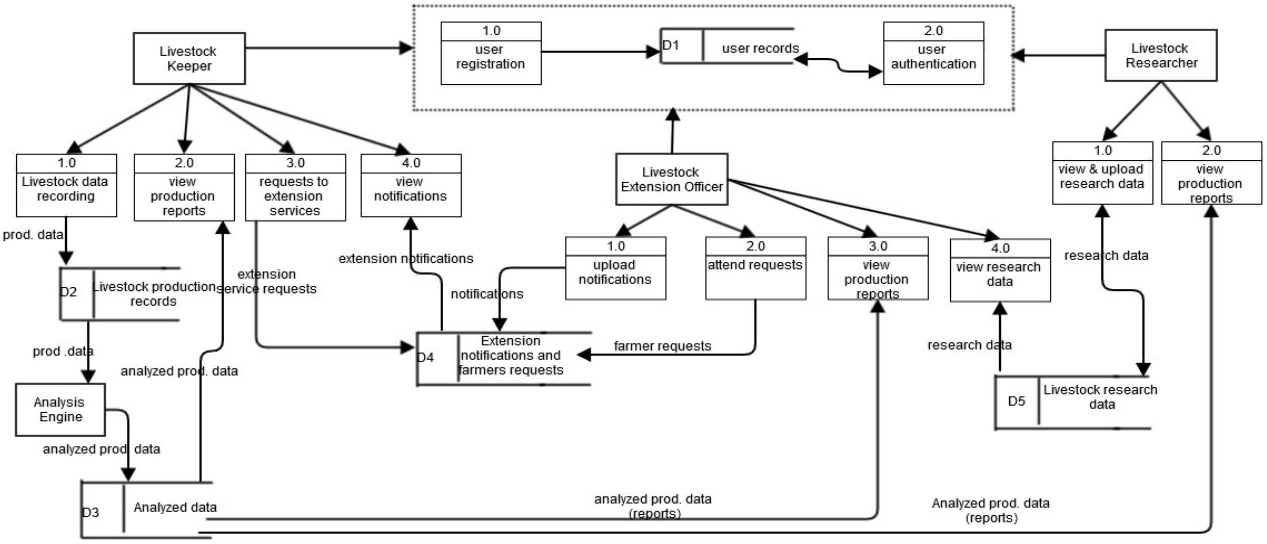
Livestock Keeper
1.0 user registration
D1 user records
2.0 user authentication
Livestock Researcher
1.0 Livestock data recording
2.0 view production reports
3.0 requests to extension services
4.0 view notifications
Livestock Extension Officer
1.0 view & upload research data
2.0 view production reports
prod. data
extension notifications
D2 Livestock production records
extension service requests
notifications
1.0 upload notifications
2.0 attend requests
3.0 view production reports
4.0 view research data
research data
prod .data
analyzed prod. data
D4 Extension notifications and farmers requests
farmer requests
research data
Analysis Engine
D5 Livestock research data
analyzed prod. data
D3 Analyzed data
analyzed prod. data (reports)
Analyzed prod. data (reports)

Figure 3. DFD level 1

❖ *Processes* - Every system has critical processes which requires protection against any significant threat. Processes of interest in the case study includes but not limited to, user registration and authentication, livestock data recording, view production reports, view requests and notifications, attending requests, upload notifications, view and upload research data, and data analysis. A security breach that might hinder the execution of any of the indicated processes implies a failure to fulfil system's functional requirements/ critical goals of the system. Furthermore, these processes are of great interest because they can be among goals of an attacker to the system.

❖ *Data flow* - The importance of protecting data on transition is as demanding as protection data at rest. The mode of data transmission from source to destination has an impact on the security of such data during transmission. Numerous applications are coded without a consideration on data protection with regard to the transmission route or when at rest on a device or system databases [10]. A breach on data flow may lead to various security threats such as information disclosure, tempering, and denial of service.

❖ *User interactions* - System and user interactions needs to be secured against security attacks in order to protect the interests of legitimate users. Applications needs to be protected from threats such as viruses, spyware and phishing to enable harmony between security and usability [11]. A security flaw can be done during system to system interaction, or user to system interaction which may lead into identity theft or other forms of man in the middle attacks.

Security breach in any of the identified resources and assets is going to be modeled using the STRIDE threats model which depends on the goals of an attacker.

C. Determination and Ranking of Threats

Threats can now be determined by following the STRIDE threats model. From the data flow diagrams, all system assets, entry points, and users have been highlighted. This will help in identifying potential threats target according to STRIDE model. In addition to DFD decomposition, a thorough description of STRIDE threats in LDC system components is indicated in Fig. 4. Decomposition of the LDC system in DFDs has enabled the realization of specific threats in each component on the system as highlighted in Fig. 4.

Mobile application clients: Mobile applications can easily be downloaded and decoded to reveal the source codes and critical functions [12]. A common decoding sequence is from .apk file package to dex bytecode (Dalvik VM) file format and then to .jar file, the later can be inspected in a Java development environment as application source codes. This enables an attacker to read and understand authentication functions, injection of harmful queries/scripts and tamper with authentication data.

Web application clients: Sensitive data in plain text may be captured by malicious users during transmission. If session IDs are not handled properly, malicious users may use them to tamper with user profiles as well as system data. Web application users may fall victims to Denial of Service attacks by attackers using them as attack zombies in distributed denial of service. This phenomenon can deny individual user service from LDC or deny other users service to LDC system.

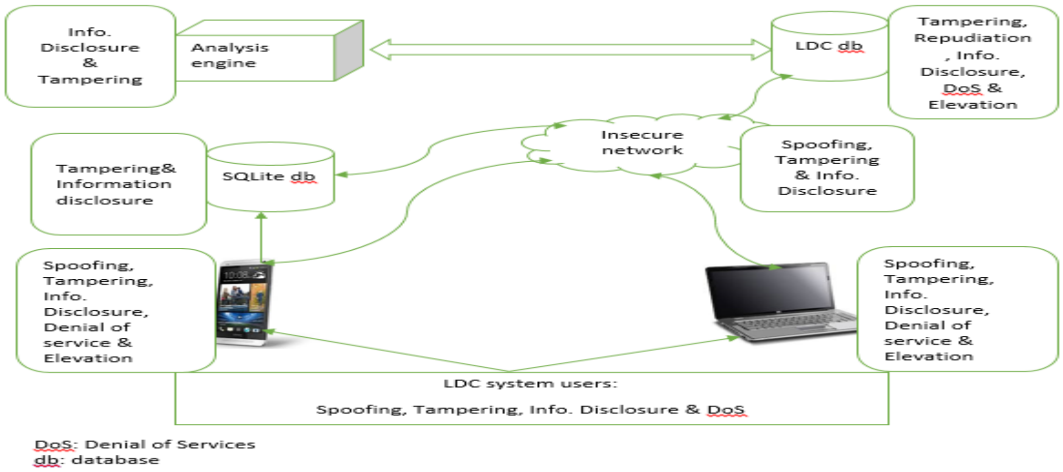
Info. Disclosure & Tampering
Analysis engine
LDC db
Tampering, Repudiation , Info. Disclosure, DoS & Elevation
Insecure network
Spoofing, Tampering & Info. Disclosure
Tampering& Information disclosure
SQLite db
Spoofing, Tampering, Info. Disclosure, Denial of service & Elevation
Spoofing, Tampering, Info. Disclosure, Denial of service & Elevation
LDC system users:
Spoofing, Tampering, Info. Disclosure & DoS
DoS: Denial of Services
db: database

Figure 4. STRIDE threats in LDC system.

SQLite database: Mobile application codes present a threat unless they are strongly typed. Study of the source code may reveal information flow, databases, table names and specific fields. Moreover, data stored temporally in SQLite databases are prone to disclosure and tampering. Studies have revealed that, android and iPhone mobile devices store a significant amount of data that can be retrieved to disclose sensitive information [13]. Data stored on an android phone can be retrieved from …/data/data directory.

Insecure network: Data on transmission is prone to a number of threats including spoofing, modification, and information disclosure [14]. Modified data during transmission may result in wrong analysis, interpretation and dissemination of incorrect data/information. Furthermore, the channel contains a huge number of third party applications and services that cannot be trusted.

LDC system database: System database is prone to client side injections such as SQL and cross site scripting which can either modify or reveal sensitive information [15]. Apart from that, improper handling of user privileges may allow an authenticated user to access unauthorized data. Untrusted applications and features in mobile devices may gain access to stored data.

Analysis engine: improper handling of server side controls may result in disclosure of procedures and functions for data analysis. Such procedures and functions may be analysis algorithms and generation of views from databases. All these can cause disclosure of sensitive data and/or modification of data and procedures.

Security threats for the livestock data center system are summarized and documented in Table III based on the system decomposition indicated in Fig. 3 and Fig. 4. The impact of all identified threats is described as either high, moderate or low. Categorization of threats impact is done based on value of the resources at risk (replacement factors and criticality to the business processes), and sensitivity of data contained in the system [16].

Threats which can have a catastrophic impact on confidentiality, integrity or availability are given a high impact factor. Catastrophic impacts may refer to the system not being able to perform its primary functions, financial loss to users of the system, and misleading information that may result into loss of life. On the other hand, a breach to security services may have a moderate impact such as reduction on effectiveness of the system to perform its primary functions. Therefore, impact categorization has no default values but rather depends on the value of the asset or resource in question, and sensitivity or criticality of such asset/resource.

TABLE III. SECURITY THREATS IDENTIFICATION USING THE STRIDE THREATS MODEL.

Threats description	Threat category	Threat impact
A malicious user may download the mobile application codes from clients' side, study authentication functions and exploit the system.	Spoofing	High
From studying the application codes, a malicious user may inject harmful queries to be executed.	Tampering with data, Information disclosure, elevation of privileges	High
A malicious user may tamper with authentication data en route from the client to the server.	Spoofing, tampering with data	High
A malicious user may tamper with data en route from clients and leads to analysis of wrong data.	Information disclosure, tampering with data	High
An attacker may flood the server with requests that may deny services to targeted users.	Denial of service	Moderate
A malicious user may obtain session ID and use it to tamper with user profiles.	information disclosure	Moderate
Malicious inputs may be used to authenticate and allow attackers to change user privileges and tamper with data.	Elevation of privileges	High
Mobile application users may download untrusted applications which through their devices may gain access to the system's server and data.	Tampering with data, Denial of service	High
A malicious user may change or delete audit logs to deny responsibility of sending or receiving data.	Repudiation	Moderate
A malicious user may redirect pages to make clients agents of attacks to other systems.	Denial of Service	Moderate
A malicious user may gain access to analyzed data and delete or modify data which will result into dissemination of wrong reports.	Tampering with data	High
A malicious user may change input validation procedures.	Tampering with data	High
A malicious user may gain access to the log file and access sensitive information that will result into security breach.	Repudiation, Tampering with data	Moderate
A malicious user may intercept data on transmission from the server to client.	Information disclosure	Low
Through steganography an attacker may insert a backdoor into a mobile client that will be able to penetrate into the system data.	Information disclosure, tampering with data	High
Due to low capability of mobile browsers, session and cookies can't easily be tracked; this may lead into exploitation of system processes and data.	Information disclosure, tampering with data, elevation of privileges	High
Not all mobile clients can be trained for secure use of applications and safe browsing.	Spoofing, information disclosure, tampering with data	High

As depicted in Table III, identified security threats in the LDC are assessed with respect to the use of web and mobile applications. This assessment is aimed at providing an overview of the major changes from other lists of threats documented, such as [17] and [18]. The use of mobile devices such as phones and tablets, raises the bar of security threats compared to the traditional web applications, for instance, Fig. 5 and Fig. 6 show that sensitive data exposure is ranked low on web applications, but, ranks top when we consider the use of mobile phones.

OWASP Top 10- 2013- The Ten Most Critical Web Applications Security risks:
A1- Injection
A2- Broken authentication and session management
A3- Cross site scripting
A4- Insecure direct object reference
A5- Security misconfiguration
A6- sensitive data exposure
A7- Missing function level access control
A8- Cross site request forgery
A9- Using components with known vulnerabilities
A10- Unvalidated redirects and forwards

The OWASP top 10- 2013 mobile application security risks:
M1- Insecure data storage
M2- Weak server side controls
M3- Insufficient transport layer protection
M4- Client side injection
M5- Poor authorization and authentication
M6- Improper session handling
M7- Security decision via untrusted inputs
M8- Side channel data leakage
M9- Broken cryptography
M10- Sensitive information disclosure

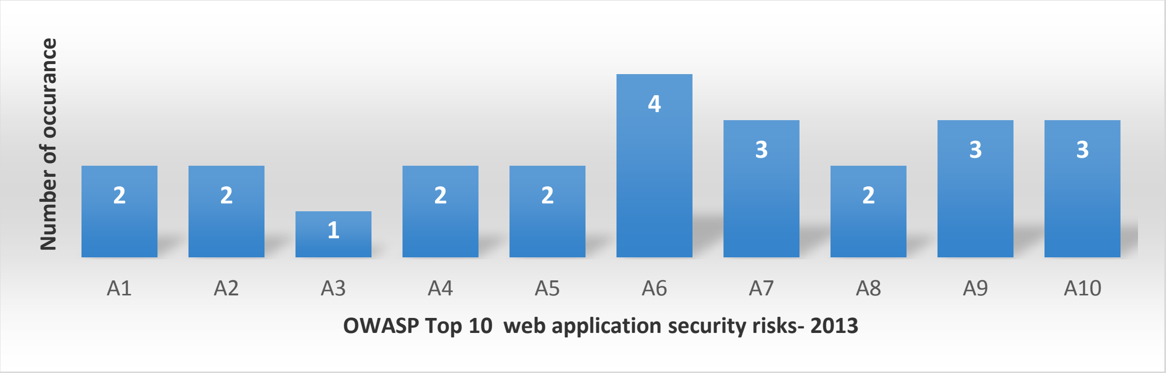

Figure 5. OWASP top 10 web application security risks occurrence in LDC system security threats identification under STRIDE threats model

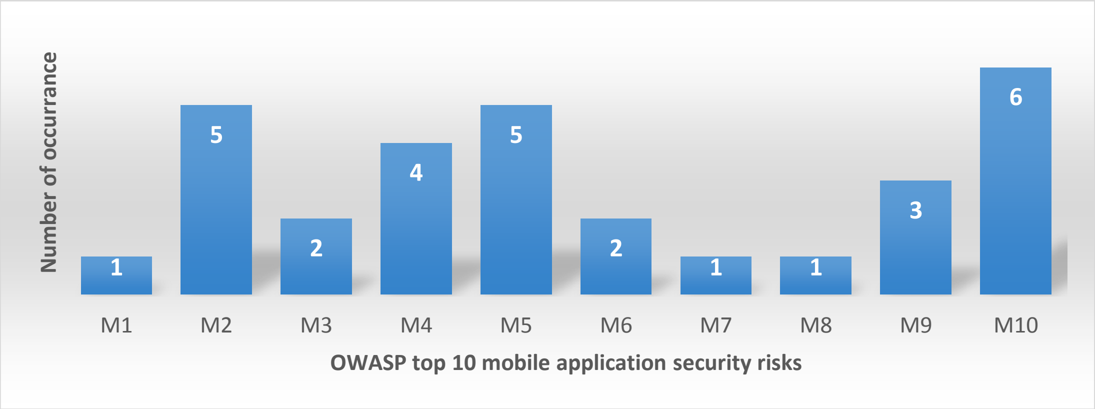

Figure 6. OWASP top 10 mobile application security risks occurrence in LDC system security threats identification under STRIDE threats model.

Fig. 5 and Fig. 6 show the critical security threats in the LDC system from analysis made using STRIDE threats model and summary results in Table III. The variation from the OWASP analysis is caused by the fact that, when we consider web applications alone, they have their own security threats which are critical; moreover, when we consider the case of mobile applications alone, they have their own critical security threats too.

The convergence of web and mobile applications implies a convergence in the security threats and formation of new threats, which can be a result of intertwined threats in specific business logic and application context. From STRIDE threats modeling results summarized in Table III, and Fig. 5 and Fig. 6, the LDC system is highly vulnerable to four prominent security threats namely: sensitive data exposure, weak server side controls, client side injection, and weak authentication and authorization. Therefore, to use web and mobile applications in one system implies a need of having a new security framework that will address security threats caused by the convergence of web and mobile applications.

3. SPECIFICATION OF THE LDC SYSTEM SECURITY REQUIREMENTS

Merged results from the STRIDE threats model, which was used in the assessment, and the threats that were ultimately identified, were used to specify security requirements for the LDC system. Documented set of security requirements are suggested by existing web applications security frameworks such as GuardRails [19], and a testing framework for web application security [20]. In addition, the specifications have also covered security suggestions provided by mobile applications security frameworks including, framework for designing, developing and using secure mobile applications [21] and Mobile applications security framework [22]. The 10 specification of the Livestock Data Center System Security Requirements are as follows:

1. All sensitive data such as passwords, profile information and production analysis should be stored or transmitted in an encrypted format. Depending on the sensitivity of data, an encryption algorithm can be selected from the simplest to the complex considering the factors of storage and transmission costs. Password protection should be done using algorithms specific for password protection such as bcrypt [23], PBKDF2 [24], or scrypt [25]. However, cryptographic hash functions such as MD5 can be an option for moderate data protection.

2. All sensitive data should not be stored in a client's device. Among the challenges of using mobile applications is connectivity, for data transferring to and from the central server. Existing options may be keeping data in a client's device until connectivity is restored, then transmission can start. Due to vulnerabilities in mobile devices such as ability to be easily put under control of attack, such practices of temporarily storing data in a client's device cannot be adequate for sensitive data.

3. Ensure the use of strong authentication functions which cannot be easily broken or understood by malicious users. Authentication functions should never be attached directly to the application codes but be used as stored procedures. In this way, it will be difficult for an attacker to get a clear picture of how a user is authenticated to use an application or a specific function. It is suggested in [22], that authentication should not only be used for users of a mobile applications but also to the devices as well. Therefore, mobile devices that accesses the LDC system should be authenticated before user is authenticated.

4. All forms collecting sensitive information should not have autocomplete options. Autocomplete option enable browsers to maintain user information which in this case, if user is entering sensitive information they can be obtained by any other. Furthermore, data fetching from a browser should not be done through a GET method (use only when necessary) since, data sent is made visible in the address bar with the use of GET method [26].

5. All functions in an application should have access control and server side authentication and authorization procedures. One way to ensure this is to check navigations in all client's pages to make sure that the pages cannot navigate to unauthorized functions.

6. The use of redirects and forwards should be avoided unless if highly demanded. If needed, ensure that the actual URL or portion of the URL is not included in the parameters. Instead, use mapping values which can be translated by the server side code. Through this, mobile application codes will also be safe from phishers who can download and study the codes.

7. Session IDs should not be written in URL. Moreover, session IDs should time out at a specific period and be rotated after a successful login. Session management on web application should be equally implemented in a

mobile application by using a cross platform language to connect to the server such as PHP.

8. Always separation of user inputs from commands/SQL statements is a good practice to avoid injection attacks. The use of bind variables is a good option to separate untrusted data from commands and queries. A good practice is to ensure that even stored procedures are implemented using bind variables. Moreover, the use of bind variables saves memory usage and make transactions faster and more scalable.

9. Mobile application users are the most vulnerable clients in the convergence of web and mobile applications. This is because, their data are more vulnerable through their devices, and also mobile browsers are vulnerable than desktop browsers. To help mobile clients have safe browsing and secure use of mobile applications, the applications should have modules that instructs the users about recommended practices and secure use of the applications to access data stored on central servers.

10. All user inputs should be validated, together with re-authentication procedures to make sure that real system users are intending to submit such input. Through this, inputs from malicious software which could penetrate the system can be avoided. Examples of such measures include the CAPTCHA [27].

4. CONCLUSION AND RECOMMENDATION FOR FUTURE RESEARCH

The convergence of web and mobile applications demand a new framework for handling security risks and controls in use. The assessment done in this paper has clearly shown the difference in threats between the use of web and mobile applications separately for specific functions as shown in Fig. 5 and Fig. 6, and a combination of web and mobile applications to deliver some specific services, as shown on Table III. The later has shown critical security concerns as security measures for web applications could not satisfy the security of a converged web-mobile application system and neither those for mobile applications.

These highlights have led into the specification of security requirements for the LDC system, which are focused on protecting the said system from security threats arising from the use of web/mobile applications. The documented requirements are not strictly focused on the LDC system, they can be adopted for any system intended to be accessed through web and mobile applications.

This paper is part of an ongoing design of a holistic security framework for the convergence of web and mobile applications. Future anticipation of the output from this paper is to be an input that will inform about important issues that the design of a security framework should entail. A live implementation of the system will be done to clearly identify most of the critical security flaws in web and mobile applications, and later design a holistic security framework for that purpose. The framework will therefore contain knowledge obtained from the implementation and together with borrowing knowledge from existing frameworks for web and mobile applications security.

ACKNOWLEDGMENT

We sincerely appreciate the support on this work from the Nelson Mandela African Institution of Science and Technology (NM-AIST) through the school of Computation and Communication Science and Engineering (CoCSE).

REFERENCES

[1] J. Lounsbury, “Application Security: From Web to Mobile. Different Vectors and New Attacks,” Security in Knowledge, 2013, pp2-30.

[2] Y. Asnar, T. Li, F. Massacci and F. Paci, “Computer Aided Threat Identification,” Requirements Engineering. Vol. 17, no. 4, 2012, pp1-8.

[3] I. Hogganvik and K. Stølen, “A Graphical approach to risk identification, motivated by empirical investigations,” SINTEF ICT and Department of Informatics, University of Oslo, 2006, P1-15.

[4] OWASP, “Application Threat Modeling,” Available: https://www.owasp.org/index.php/Application_Threat_Modeling, 2013. Last accessed 9th Jan 2014.

[5] M. Abi-Antoun and J. Barnes, “STRIDE-based security model in Acme,” CMU-ISR-10-106, 2010, pp1-16.

[6] R. Scandariato, K. Wuyts, and W. Joosen, “A descriptive study of Microsoft’s threat modeling technique,” Requirements Engineering, 2014, pp1-18.

[7] T. Xin and B. Xiaofang, “Online Banking Security Analysis based on STRIDE Threat Model,” International Journal of Security and Its Applications. Vol. 8, no. 2, 2014, pp271-282.

[8] D. Chappel, “INTRODUCING ODATA: Data access for the web, the cloud, mobile devices, and more. White Paper, 2011, pp6-10

[9] Ernest and Young, “Mobile Device Security: Understading Vulnerabilities and managing Risks,” Insights on Governance, Risks and Compliance 2012, pp2-9.

[10] J. Khonstamm, “Article 29 data protection working party,” 00461/13/EN WP 202, 2013, pp5-12.

[11] K. P. Yee, “Guidelines and strategies for secure interaction design,” Security and Usability: Designing Secure Systems That People Can Use, 2005, pp247-273.

[12] W. Enck, D. Octeau, P. McDaniel and S. Chaudhuri, “A Study of Android Application Security,” In USENIX Security Symposium, August, 2011.

[13] N. Al Mutawa, I. Baggili, and A. Marrington, “Forensic analysis of social networking applications on mobile devices,” Digital Investigation, Vol. 9, 2012, pp24-33.

[14] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, Vol. 34, no. 1, 2011, pp1-11.

[15] H. Yang, and N. Zhihong, "A database security testing scheme of web application," Computer Science & Education, 2009. ICCSE'09. 4th International Conference on. IEEE, 2009.

[16] D. P. Duggan, S. R. Thomas, C. K. K. Veitch, and L. Woodard, “Categorizing threat: Building and using a generic threat matrix,” Sandia National Laboratories report SAND2007-5791, Albuquerque, New Mexico, September, 2007.

[17] OWASP, “OWASP Top 10- 2013. The Ten Most Critical Web Application Security Risks,” The Open Web Application Security Project, 2013, pp1-22.

[18] OWASP, “Top 10 Mobile Risks, Release Candidate v1.0.,” Available: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_Ten_Mobile_Risks,2013, Last accessed 17th Jan 2013.

[19] J. Burket, P. Mutchler, M. Weaver, M. Zaveri and D. Evans, “GuardRails: a data-centric web application security framework,” In Proceedings of the 2nd USENIX conference on Web application development, USENIX Association, June, 2011, pp. 1-1.

[20] Y. W. Huang, C. H. Tsai, T. P. Lin, S. K. Huang, D. T. Lee, and S. Y. Kuo, “A testing framework for Web application security assessment,” Computer Networks, Vol.48, no.5, 2005, pp739-761.

[21] M. A. Serhani, A. Benharref, R. Dssouli, and R. Mizouni, “Toward an Efficient Framework for Designing, Developing, and Using Secure Mobile Applications,” Proceedings of World Academy of Science: Engineering & Technology, 2009, pp.52.

[22] K. P. Yadav and R. Mishra, “Mobile Application Security Framework,” IT Best Practices Alert, Network World. 2013, pp.1-5.

[23] T. De Raadt, N. Hallqvist, A. Grabowski, A. D. Keromytis and N. Provos, “Cryptography in OpenBSD: An Overview,” In USENIX Annual Technical Conference, FREENIX Track. June, 1999, pp.93-101.

[24] Y. S. Dandass, “Using FPGAs to parallelize dictionary attacks for password cracking,” In Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, 2008, pp. 485-485.

[25] C. Percival and S. Josefsson, “The scrypt Password-Based Key Derivation Function,” 2012.

[26] V. Sahgal, “In HTML forms, what’s the difference between using the GET method versus POST?,” Available: http://www.programmerinterview.com/index.php/general-miscellaneous/html-get-vs-post/. Last accessed 28th Feb 2013.

[27] M. Sanghavi, and D. Shreyas, "Progressive captcha," U.S. Patent Application 11/929,716, filed October 30, 2007.

Ms. Devotha Nyambo is a holder of BSc. in Computer Science and currently she is a master’s student at the Nelson Mandela African Institution of Science and Technology (NM-AIST), Tanzania. She is pursuing a MSc. degree in Information and Communication Science and Engineering foremost Information Systems Development and Management. Devotha is passionate about information systems security modeling, specifically mobile-web applications.

Eng. Dr. Zaipuna O. Yon[redacted]ah MIET, MIEEE - holds a B.Sc. degree (with Hons - 1985) in Electrical Engineering from University of Dar es Salaam - Tanzania; and M.Sc. (1986) and PhD (1994) Degrees in Computer-Based Instrumentation and Control Engineering from the University of Saskatchewan, Saskatoon - Canada. In Tanzania, he is a Registered Consulting Engineer in ICTs. Dr. Yonah has over 30 years of practice. His work spans the academia, industry and policy making fields. He is currently associated with The Nelson Mandela Institution of Science and Technology – (school of Computation and Communication Science and Engineering), and the IEEE Inc.

He is one of the pioneers driving the national broadband agenda in Tanzania. He believes that ICTs, as tools for development, promise so much: *interactivity, permanent availability, global reach, reduced per unit transaction costs, creates increased productivity and value, jobs and wealth, multiple source of information and knowledge*. Armed with such a belief, his current work aims at creating and delivering value through ICT-enabled services in the shortest times possible. His research interests include: ICT4D, Cyber Security, ICT Policy and Regulation, Mobile and Web applications, high-capacity broadband networks, Intelligent Instrumentation and Control Engineering; and ICT enabled 21st Century Education delivery (ICT4E).

Dr. Charles N. Tarimo is an active researcher on ICT security issues, with research interests focused on operational and practical issues with regard to aspects of security requirements development, designing, implementation, and management of different technical and non-technical ICT security controls within organizations/enterprises as well as research on similar issues at the national level.

He has been collaboratively working with other researchers to carry out different research studies in the field of Information and Communication Security and published the research findings at various International Conferences. Dr. Tarimo is currently an employee of the University of Dar es Salaam, working at the College of Engineering and Technology, serving as the University's ICT Manager. But also he is involved in the teaching of related subjects in computer engineering, such as computer hardware and software engineering, computer and networks security, computer networking as well as artificial intelligence.