International Journal of Scientific Engineering and Science
Volume 2, Issue 1, pp. 52-56, 2018. ISSN (Online): 2456-7361

Web Request Level Protection of Cyber Applications against Threats and Attacks

[1]BELLO Rotimi-Williams, [2]MORADEYO Oluwatomilola Motunrayo, [3]OLANIYAN Abolade Shekinat

[1]Doctorate Student, Department of Computer Science, Federal University of Technology, Akure, Ondo State, Nigeria
[2, 3]Computer Science Department, The Ibarapa Polytechnic, Eruwa, Oyo State, Nigeria

Abstract— *One of the risks of cyber technology is the threat that comes along its benefits: by the very nature of the opportunities it presents it becomes a target for cyberattacks, industrial espionage, and cybercrime. Therefore, this paper intended to advance its protection as a paramount priority by devising aggressive techniques that can prevent and militate against its threats and attacks. This is achieved by designing defence architecture with security check point to detect the web-request-level attacks by employing request blocker and Rivest, Shamir, and Adleman (RSA) algorithm. The results in this paper showed that the cyber application which is made up of web controls the access over its functions by checking session variables indicating the user privilege before its restrictive functions can be executed. If the application is not at the required state, the web will redirect the user to the login page, authorization page or an error page. Sensitive and security information about the activities of cyber threats was revealed and prevented.*

Keywords— *Espionage, Blocker, Cybercrime, RSA, and Cyberattack.*

I. Introduction

As technology advance throughout history, new opportunities are created, thereby bringing about those that exploit them for their own gain. Despite the threat of viruses and malware almost since the dawn of computing, awareness of the security and sanctity of data with computer systems didn't gain traction until the explosive growth of the internet, whereby the exposure of so many machines on the web provided a veritable playground for hackers to test their skills – bringing down websites, stealing data, or committing fraud. It's something we now call cybercrime. Since then and with internet penetration globally, the opportunities for cybercrime have ballooned exponentially. Combating this is a multi-disciplinary affair that spans hardware and software through to policy and people – all of it aimed at both preventing cybercrime occurring in the first place or minimising its impact when it does [1]. This is the practice of cybersecurity as simply illustrated in Fig. 1. However, cybersecurity is a constantly evolving, constantly active process just like the threats it aims to prevent. As we integrate technology further into our lives, the opportunities for abuse grow. So too, then, must the defences we employ to stop them through the education, awareness and practice of cybersecurity. It is in literatures that the first web-request-level attack is authentication/authorization (auth) bypass [2-4]. The web application controls the access over its functions by checking session variables indicating the user privilege before its restrictive functions can be executed. If the application is not at the required state, the web application will redirect the user to the login page, authorization page or an error page. However, if there exists a path leading to the restrictive function with insufficient or erroneous checking of session variables, the attacker is able to bypass the authentication/authorization.

The second web-request-level attack is parameter manipulation [2], [3], and [5]. In a lot of cases, the web application system assumes implicit relations between the user's input parameters within web requests and the session state. Such a relationship may also be reflected from web responses returned by the web application. If the application doesn't check the session state when accepting the web request, the attacker is able to manipulate the input parameters and gain access to unauthorized information. The third web-request-level attack is workflow bypass [6]. A web application usually has an intended workflow, which requires the user to perform a predefined sequence of operations to complete a certain task. Therefore, this research work intended to advance cyber's protection as a paramount priority by devising aggressive techniques that can prevent and militate against its threats and attacks. The security specifications approach used in this research work to great extent prevent session variable/parameter manipulations and authorization/authentication flaws. It also revealed sensitive and security information about the activities of cyber threats.

The following three types of vulnerabilities that are possibly introduced into web application are captured via a general specification based on information flow models [6]: (a) insufficient definition of session variables for differentiating all possible states. (b) Insufficient checking of session variables at appropriate program points. (c) Erroneous checking of session variables that can be bypassed. Due to the stateless nature of HTTP protocol, session variables are explicitly defined in web application to maintain the state of a web session. There are two ways for maintaining session states [7]: (a) Client side only: Where session states are directly carried in cookies, hidden forms, or URLs. (b) Collaboration of the client and the server: where the server stores the session states and issues a session ID to the client for indexing its session states. In either case, session states can be retrieved at runtime for each web request independent of the web application system implementation. For example, when session states are carried in cookies, hidden forms, or rewritten URLs, they can be directly retrieved from the web requests. When session states are kept in the server side, they can be found either in a file or a database table.

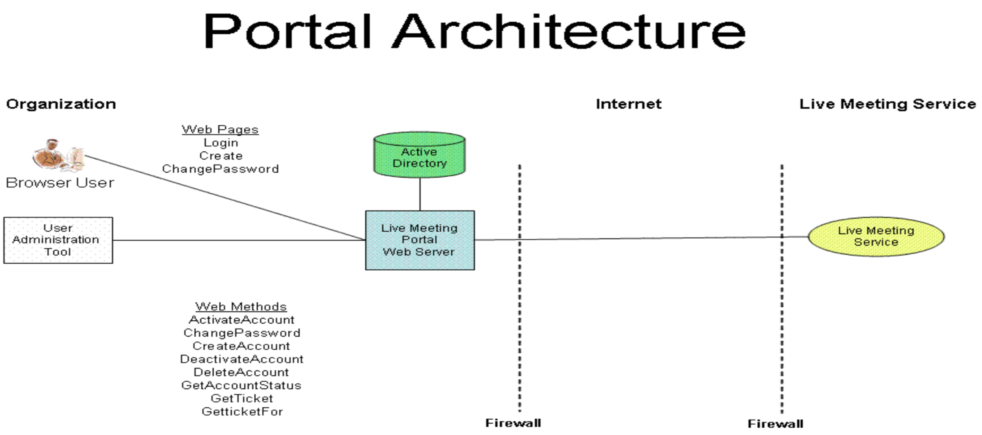


Fig. 1. A simple secured portal platform.

II. Methodology

The proposed system is based on the black-box approach (BLOCK) for detection of state violation attacks (Fig. 2). This proposed system has request-blocker for security-check-point. The request-blocker, which sits between the users and the web application, operates over a central decision mechanism based on repository of security specifications (rules), and from the interactions between the web and web clients, database and other components, the security specifications (rules) are inferred. The decision mechanism has access to the user session information through connectors which can read local session files or retrieve information from database table; this is capable of evaluating the request and queries in a context-aware manner and identifying logic attacks (Fig. 3).

Fermat's theorem, Euler's theorem and Euler's totient function were employed as the constituents of RSA algorithm in the web development. RSA algorithm, conceived by Rivest, Shamir and Adleman (RSA) is a mathematical model that is largely synonymous with cryptography for encryption and decryption. The materials used for this research work is the clinical procedure manual. The extracted data and information from the manual formed the field names used in the web applications design. For the implementation, the web applications are deployed on 2.0 GHz Core 2 SQL server with 2GB on 64 bit running Windows 8 OS and Microsoft Visual Studio 2010 Ultimate VB.NET programming language. The evaluation of the proposed system is made by comparing the results obtained with Open-Electronic Medical Record (Open-EMR), one of the most popular open source electronic health records and medical practice management applications, which is known to contain a number of security vulnerabilities [8]

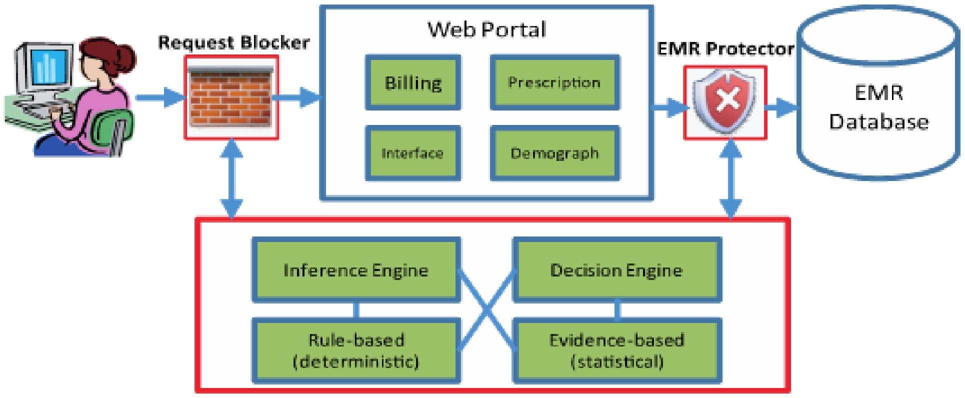


Fig. 2. Two-tier defence architecture for web application (Li and Xue, 2012).

RSA algorithm for the security specifications (rules) is as follows

Deciphering an enciphered message gives the original message as shown in (1)

$$D\ (E\ (M))=M \quad (1)$$

Reversing the procedures still returns M as shown in (2)

$$E\ (D\ (M))=M \quad (2)$$

where,

D is the decryption procedures, *E* is the encryption procedures and *M* is the message.

for the encryption procedures,

$$C = P^{e} \bmod M \quad (3)$$

where,

C is the ASCII value of a byte of cipher text, *P* is the plain text, *e* is the encryption key and *M* is a value called the modulus.

for the decryption procedures,

$$P = C^d \bmod M \quad (4)$$

where,

d is the decryption key

Fermat's theorem for the RSA algorithm is as follows:

$$M^{p-1} \equiv 1 \bmod p \quad (5)$$

where,

M is any positive integer not divisible by any prime number *p*.

Euler's theorem for the RSA algorithm is as follows:

$$M^{\varphi(n)} \equiv 1 \bmod n \quad (6)$$

where,

M and *n* are any two integers that are relatively prime, $\varphi(n)$is the number of integers less than *n* and relatively prime to *n*.

Euler's totient function for the RSA algorithm is as follows:

$$\varphi(n) = \varphi(pq) = \varphi(p) \cdot \varphi(q) = (p-1) \cdot (q-1) \quad (7)$$

where,

p and *q* are any two prime numbers and *n* is their product, $\varphi(n)$ is the number of integers less than *n* and relatively prime to *n*.

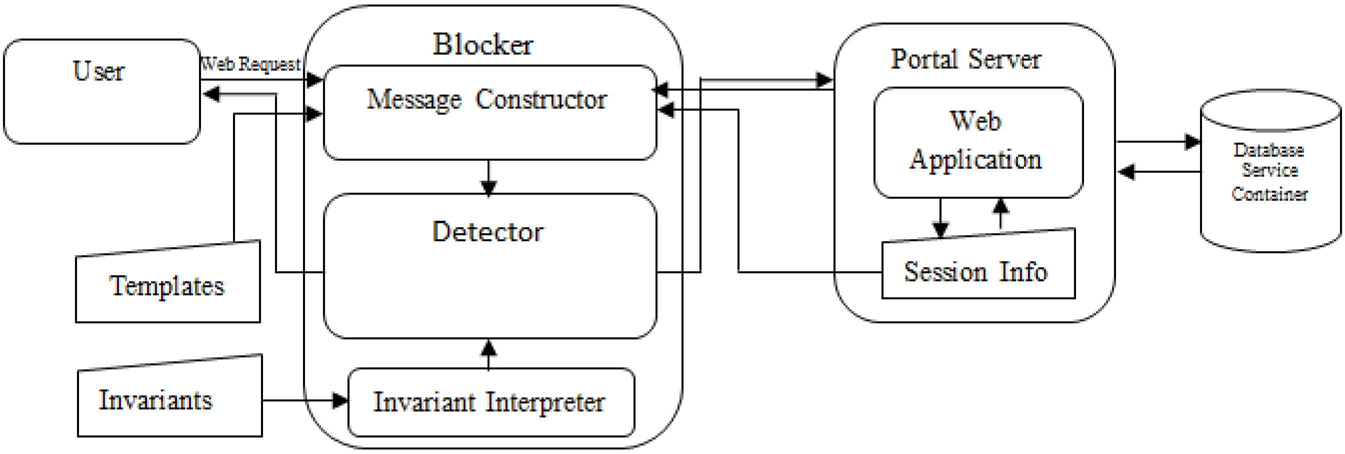


Fig. 3. Architecture of a secured web application with security check point.

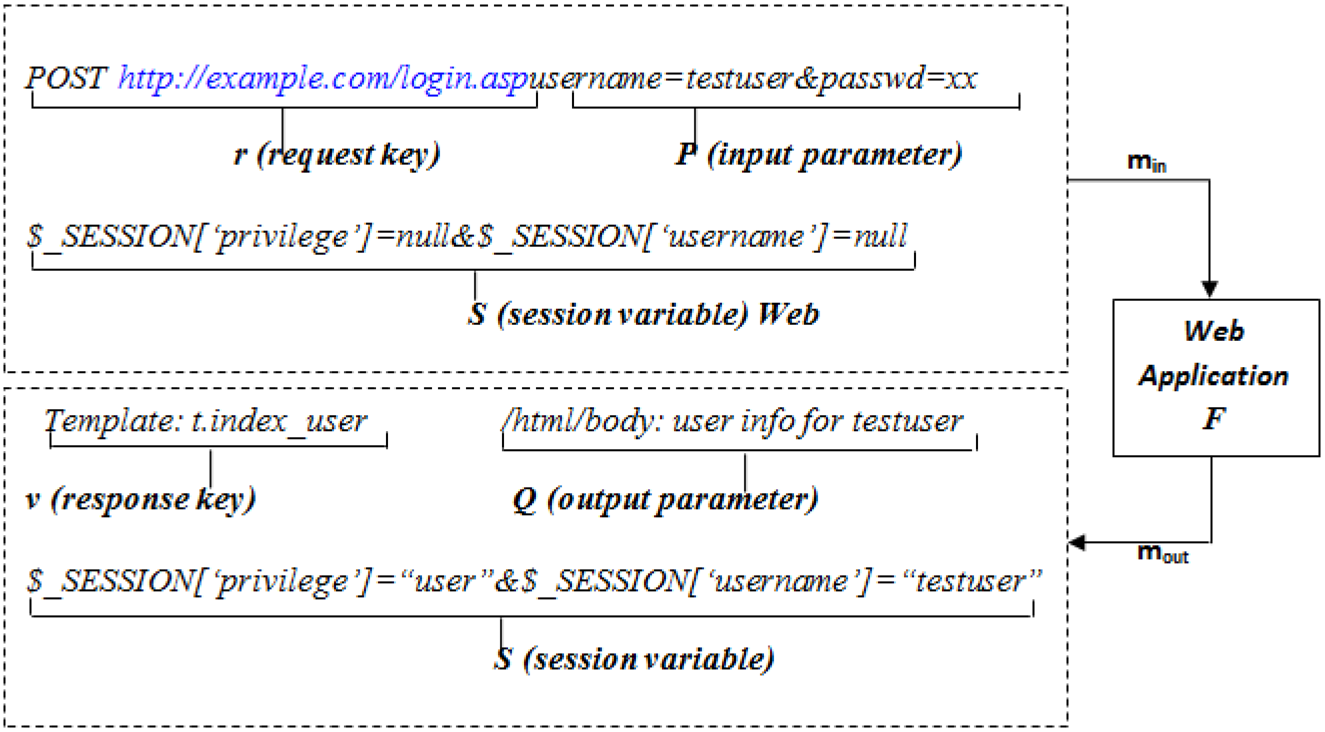

POST http://example.com/login.aspusername=testuser&passwd=xx
r (request key)
P (input parameter)
$_SESSION['privilege']=null&$_SESSION['username']=null
S (session variable) Web
min
Web Application F
mout
Template: t.index_user
/html/body: user info for testuser
v (response key)
Q (output parameter)
$_SESSION['privilege']="user"&$_SESSION['username']="testuser"
S (session variable)

Fig. 4. A stateless view of web application.

2.1 Formalization of a Web Application as a Stateless System

Fig. 4 is a stateless view of web application. A web application is regarded as a stateless system *F*, which accepts an input m_{in} and releases an output m_{out}, expressed as $F(m_{in}) = m_{out}$. An input m_{in} consists of a web request and a set of session variable name/value pair $S(m_{in})$. To facilitate detection, a web request is further decomposed into two components:

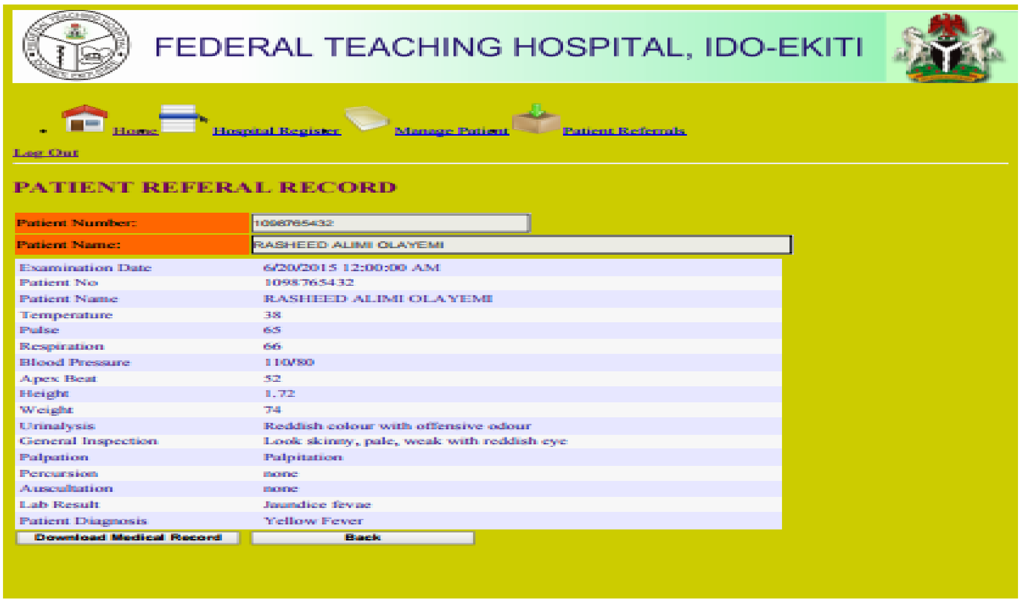
a. Web request key $r(m_{in})$, which includes the HTTP request method and the target file, and a set of input parameter name/value pair $P(m_{in})$.

b. Similarly, an output consists of a web response and a set of session variable name/value pair $S(m_{out})$.

A web response is a synthesized web page, which is usually generated by filling dynamic contents into static web page structure (i.e., template). To deal with the infinite number of possible web responses, a web page is decomposed into a web template, the number of which is finite, with a set of dynamic contents, which become output parameters. If a unique ID is assigned to each static template, a web response can be symbolized as a web template ID (i.e., web response key $v(m_{out})$) and a set of output parameter name/value pair $Q(m_{in})$.

III. Results and Discussions

Fig. 5 is an implemented snapshot of web based patient referral record among the patient's health handlers. The extracted data and information from the clinical procedure manual were used in the programming module for basic health related information on the health web. The programming module coupled with the system model ensures utmost protection of patient's data and information during online referral against web-request-level attacks and vulnerabilities mentioned in the introductory part of this paper.

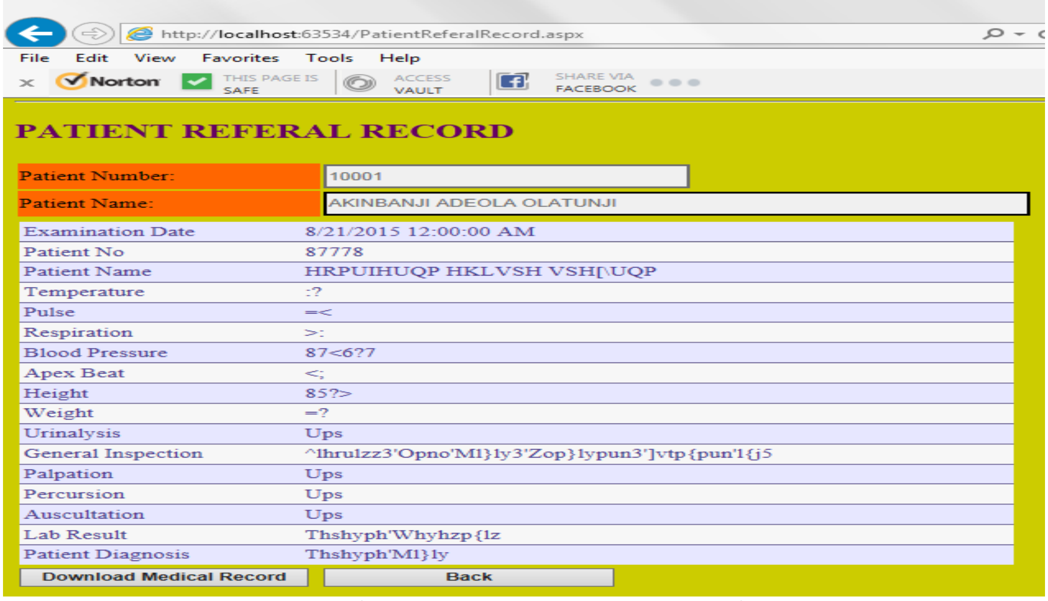


PATIENT REFERAL RECORD

Patient Number:	1098765432
Patient Name:	RASHEED ALIMI OLAYEMI

Examination Date	6/20/2015 12:00:00 AM
Patient No	1098765432
Patient Name	RASHEED ALIMI OLAYEMI
Temperature	38
Pulse	65
Respiration	66
Blood Pressure	110/80
Apex Beat	52
Height	1.72
Weight	74
Urinalysis	Reddish colour with offensive odour
General Inspection	Look skinny, pale, weak with reddish eye
Palpation	Palpitation
Percursion	none
Auscultation	none
Lab Result	Jaundice fevae
Patient Diagnosis	Yellow Fever

Fig. 5. Snapshot of download able patient medical referral record.



PATIENT REFERAL RECORD

Patient Number:	10001
Patient Name:	AKINBANJI ADEOLA OLATUNJI

Examination Date	8/21/2015 12:00:00 AM
Patient No	87778
Patient Name	HRPUIHUQP HKLVSH VSH[\UQP
Temperature	:?
Pulse	=<
Respiration	>:
Blood Pressure	87<6?7
Apex Beat	<;
Height	85?>
Weight	=?
Urinalysis	Ups
General Inspection	^lhrulzz3'Opno'Ml}ly3'Zop}lypun3']vtp{pun'l{j5
Palpation	Ups
Percursion	Ups
Auscultation	Ups
Lab Result	Thshyph'Whyhzp{lz
Patient Diagnosis	Thshyph'Ml}ly

Fig. 6. Snapshot of encrypted downloadable patient medical referral record.

The system mechanism is as following. (i) It validates the input: the web request is accepted, if and only if the request key has been observed and all the invariants associated with it are satisfied. Otherwise, the web request is dropped. ii) It validates the input/output pair: The web page is sent back to the user if and only if the corresponding key pair has been observed and all the invariants associated with it are satisfied. Otherwise, the web page is blocked. All the intending attacks are detected when: (a) each auth bypass attack instance violates the invariants associated with request keys and is detected. (b) The parameter manipulation attack violates the invariant associated with the request key where the input parameter username is always equal to the session variable and is detected. (c) The workflow bypass attack violates the invariant associated with the request key that the key pair always precedes the request key and is detected (Fig. 6).

IV. Conclusion

The security approach used in this research work is valuable in that it is independent of the web application source code and suitable for a large variety of web application hosting scenarios based on different application frameworks, where the source code may not be available. Visual Basic.NET, the programming language that was used for the implementation possesses some characteristics and capability that underline the decision to use it. The rich and extensive API enables developers to easily incorporate packages into their codes which enhance reusability and utility of codes. The following are some of the additional advantages of the programming language used. (a) Code can be created faster and better because there is no need to create functionality from scratch. Best of all, many developers can create their own packages and make them open-source, allowing other developers to use their codes and hopefully make improvement. (b) The packages are well documented, so learning about a particular method is as easy as incorporating them in one’s code, a phenomenon that dramatically improves programmer’s productivity. (c) The major problems in programs like memory management and bounds checking are not seen in VB.NET, all these issues are handled automatically. (d) Serial I/O, networking, graphics sound, and even video are all supported packages. (e) Most importantly, administrators’ passwords and other sensitive details resident in database are well encrypted against manipulations. Also, many sites from the last decade are static, but more and more people are realizing the advantages of having a dynamic portal, among which are: (i) It possesses some security features (ii) it is much easier to update (iii) it is much more functional website (iv) its new content brings people back to the site and helps in the search engines (v) it can work as a system to allow staff or users to collaborate. Although, to develop a secured web application is a complex and challenging task because the development adds additional layer of complexity during implementation. The web application has to implement and enforce complex security policies to restrict the access of sensitive information and actions such as diagnosis code and secret tokens against common security pitfalls. This complexity is increased because of the concurrent nature of the internet, and the interplay among many users. Such policies are usually dynamic, related with clinical workflows, thus cannot be explicitly or precisely defined. Moreover, the inclusion of business logic in the models would allow for more complex model checking techniques to be applied to ensure even more secure implementations of the modelled system. Finally, the reflection of a secured web-based healthcare e-referral capability of the research makes it suitable for any healthcare institutions and this will greatly reduce patient referral problems.

References

[1] Australia Computer Society, Cybersecurity guide, Level 11, 50 Carrington Street, Sydney, 2016.

[2] X. Li and Y. Xue, “Block: A black-box approach for detection of state violation attacks towards web applications,” In *ACSAC ’11*, pp. 247–256, 2011.

[3] X. Li and Y. Xue, “Sentinel: Securing database from logic flaws in web applications,” In *CODASPY ’12*, pp. 25–36, 2012.

[4] V. Felmetsger, L. Cavedon, C. Kruegel, and G. Vigna, “Towards automated detection of logic vulnerabilities in web applications,” In *USENIX’10: Proceedings of the 19th Conference on USENIX Security Symposium*, pages 143-160, 2010.

[5] S. Chen, R. Wang, X. Wang, and K. Zhang, “Side-channel leaks in web applications: A reality today, a challenge tomorrow,” In *Oakland 10*, pp. 191–206, 2010.

[6] R. Message, and A. Mycroft, “Controlling control flow in web applications,” *Electron. Notes Theor. Comput. Sci.200*, pp. 119–131, 2008.

[7] R. Sekar, “An efficient black-box technique for defeating web application attacks,” In *NDSS’09. 16th Annual Network and Distributed System Security Symposium*, 2009.

[8] Y. Hong, S. Lu, Q. Liu, L. Wang, and R. Dssouli, “Securing telehealth applications in a web-based e-health portal,” *IEEE Proc. 3rd International Conference on Availability, Reliability, and Security (ARES 2008)*, 2008.