A Model for Security Evaluation of Digital Libraries: A Case Study on a Cybersecurity Curriculum Library

Nnatubemugo "Ugo" Ngwum[1]
nngwum1@students.towson.edu

Sagar Raina[2]
sagar.raina@msmc.edu

Sabina Aguon[3]
saguon1@students.towson.edu

Siddharth Kaza[4]
skaza@towson.edu

Towson University [1,3,4]
7800 York Road, Towson, MD 21252, USA

Mount Saint Mary College[2]
330 Powell Ave., Newburgh, New York, 12550, USA

Abstract - The use of digital libraries (DLs) is increasing. To attract users and sustain digital libraries, security of these systems is critical. Through extensive review of literature, standards and other security technical reports, we propose a model for security evaluation of digital libraries and test the effectiveness of the model using the CLARK cybersecurity curriculum digital library (www.clark.center) at Towson University. We identify five core security criteria that are broken down into several security requirements that a DL should fulfil to achieve security. Results from the evaluation, which include static code analysis and expert review of CLARK's security mechanisms, indicate the proposed model is significantly effective in evaluating the security requirements of digital libraries.

Keywords

Security evaluation of digital libraries (DLs), security evaluation, security metrics

1. INTRODUCTION

A digital library (DL) is a complex information system that holds and manages digital content. With the convenience, cost-effectiveness and ability to access digital content from anywhere, DLs facilitate knowledge creation and dissemination, influencing its increasing adoption. Several DL projects across domains have been undertaken, seeking to provide various services to its users/clients [6].

Considering the increasing use of digital libraries, there is need for evaluation of these systems to probe for challenges and limitations that could deter their use and large-scale adoption by their target audiences. Several studies have addressed DL evaluation from different perspectives, ranging from user-centered

[1,2,3], system-centered [4] to impact analysis of DL usage [5] in various fields of study. However, these evaluation efforts have subtly addressed or totally ignored the security aspect of DLs.

To attract and retain an active user base to achieve their goal(s), DL systems must be evaluated to address security issues, as security is a critical concern of any information system, which cannot be neglected. Therefore, in this paper, we address the following research questions:

RQ1: Are there effective models or tools for security evaluation of digital libraries?
RQ2: What components should be included in the security evaluation of a DL?

To address these questions, we develop a model for security evaluation of digital libraries, and further break it down into specific security requirements in the evaluation checklist used. We then use the model to evaluate CLARK digital library to test its effectiveness in assessing other DLs.

Following sections present a review of existing evaluation efforts in Section 2, research framework and proposed evaluation model in Section 3, results and findings in Section 4, and conclusion and future direction in Section 5.

2. BACKGROUND

Over the years, several research studies have evaluated DL systems: user-centered evaluations, usability assessment, impact analysis, and system-centered studies [13]. Very few studies have focused on security evaluation. We first present digital library evaluation studies, and then studies focused on security evaluation.

Digital Library Evaluation Studies

Blandford et al. [3] proposed a framework for planning and conducting DL evaluation with a user-centric approach that focuses on user-system interactions. Similarly, Tsakonas et al. [2] explored the interaction of the various components of a DL (user, collection and system) and the dynamic relationship they share as a cohesive whole. Bertot et al. [1] adopted a multi-method approach for evaluating DLs. These user-centric studies attempt to capture functionality, usability and accessibility testing, with little or no emphasis on system security. Saracevic et al. introduced the foundational framework for evaluating digital libraries, enumerating four elements – context, construct, criteria, measures and methodology – which any evaluation study should consider [8]. Several studies have used this framework as a structural basis for their study. However, no tool or clear specification on each of those elements was given by their framework in terms of what exactly should be investigated in a system. Nicholson [4] claims a holistic approach by viewing the evaluation from four different quadrants: the internal view, which compared system's component against standard; the external view, which focused on the system results; the external view of use, focused on how the results are valued; and the internal view of use that examined the interactions between the technical components.

With few studies focusing on both user-centered and system-centered considerations [4,8], others remain within usability confines [9]. Besides usability assessments, understanding the actual impact of DL usage on users' learning outcomes has also been explored [5].

Security-related evaluations in digital libraries have been captured by few studies as a part of regular technical evaluation [10, 11]. None of the studies has offered a dedicated security-centered approach or model for evaluating and ensuring all-round security of digital libraries. A digital library is a software system with several components including front-end application, collection or database and back-end servers and functionality mechanisms. Wang et al. [11] defines a set of security metrics for evaluating software systems. Their work only focuses on the quantitative rating of software vulnerabilities.

In this work, we advocate security in digital library as an individual component, and adopt a well-received evaluation framework by Saracevic [8] as our research methodology to propose a model for evaluating and securing DLs.

3.0 RESEARCH METHODOLOGY

The DL evaluation framework by Saracevic [8], which we adopt as our methodology for this study, outlines the key elements or areas any digital library evaluation study should address. These elements include: (1) the context, which explains the goal or focus of the evaluation (e.g., usability, impact analysis, technology, security, etc.); (2) construct, which defines the exact components or parts of the system to be evaluated; (3) criteria, which defines parameters of performance; and (4) methodology, which describes the measures, instruments and approach for conducting the evaluation. In subsection 3.1, we discuss these elements for our study.

3.1 Context

The goal of this study is to address security of a digital library. This involves ensuring confidentiality, integrity and availability of information assets. Applying mechanisms to attain these security attributes prevents unauthorized access to digital resources, protects data from unauthorized modification, and ensures that resources are always accessible to only authorized users [12] respectively. Based on this context, we hypothesize that the security of a digital library will be achieved when each of the DL components as defined by Tsakonas (user, collection, system) satisfies confidentiality, integrity and availability attributes.

3.2 Construct

A typical digital library consists of three core components including user, system and collection [2]. These are the main components without which the DL does not exist; hence, securing them would be required to secure the entire DL system [14].

User component focuses on user interaction with digital library using application interface. This interaction could be used as a possible attack surface if the user behavior is not restricted. Examples include the interface allowing invalid inputs, interface not warning users of unsafe actions, GUI storing sensitive information in clear text and so on.

System component includes all hardware and software that enable the overall functionality of the digital library (e.g., servers, platforms, programming frameworks and libraries, security architectural approach etc.). As users request information from a DL, the entities in this component interact, process the request and return the results to the user. The interaction between these entities is another possible attack surface that could be exploited. Therefore, adequate security measures are required within the entities of the system component to prevent security breaches.

The collection component includes the digital library database, one of the critical resources of the DL system. DL database stores the user data and the content. One of the goals of the attackers is to target the system database. Therefore, security of the DL database becomes critical. Appropriate measure (described in criteria section) should be put in place to prevent any database breaches.

3.3 Criteria

To evaluate the security of a digital library, we identify five key security criteria's including: 1) Encryption, 2) Authentication and Authorization, 3) Platform weakness and vulnerabilities, 4) System and Security Audit, and, 5) Usability and Human-factor. These are further broken down into specific security requirements (Table 3.2).

3.3.1 Encryption Mechanism

Cryptography involves conversion of data to forms unreadable to third party (who can be potential adversary) and re-conversation of same data to the original readable form at the receiving end or system using a secret key. Encryption ensures confidentiality of the data. Under this criterion, we identify DL components (user, system or collection) where implementing cryptographic mechanisms are necessary. In addition, we assess cryptographic tools and strategies that are deployed in DL systems for adequacy and alignment with security standards.

3.3.2 Authentication and Authorization Mechanism

Authentication and Authorization of users in an information system is a critical security measure. In order to ensure security of a digital library, correct controls for granting and controlling/managing access between user-system and system-system interactions must be implemented. This criterion assesses the standard authentication and access control mechanisms of digital libraries.

3.3.3 Platform Weaknesses and Vulnerabilities

Majority of the cyber attacks are attributed to insecure software development [21]. In addition, recent system/software development trends reveal that programmers are increasingly leaning towards use of existing frameworks and libraries rather than developing code from scratch [22]. Often, these reusable components exhibit inherent vulnerabilities that are potentially transferred to the program or system in which they are used. Therefore, it is important to identify and assess vulnerabilities in DL components that may impose a significant threat to the system.

3.3.4 System/Security Audit

Logging system events, both legitimate activities and attack exposures, are critical security measures to identify significant system intrusions. In this criterion, we identify standard requirements for system events logging, log records management responsibilities, and action plans for identified security events.

3.3.5 Usability and Human-Factor Support

Usability is concerned with how easy-to-use a system's interface or website is, while human factor considerations is a broader term which seeks to address and limit inappropriate and risky user behaviors by means of adequate security mechanisms. A popular perception in technology industry is that "the more the security, the less usable a system". While this perception may hold for several scenarios, we think through to identify that for a system's security features to be effective, they have to "lend themselves to be easily configured and used" [15]. Not designing the system's architecture, features, data flow, and especially user interfaces, such that they support security and reduce risky user behaviors would increase system's exposure to security incidents. Furthermore, a system not being usable would naturally deter usage and affect its massive adoptions due to navigation challenges limiting its availability. Moreover, availability is a key security attribute that must be ensured.

Figure 3.1 represents our model, with three merging sectors that represent the three DL components that come together to form the system.

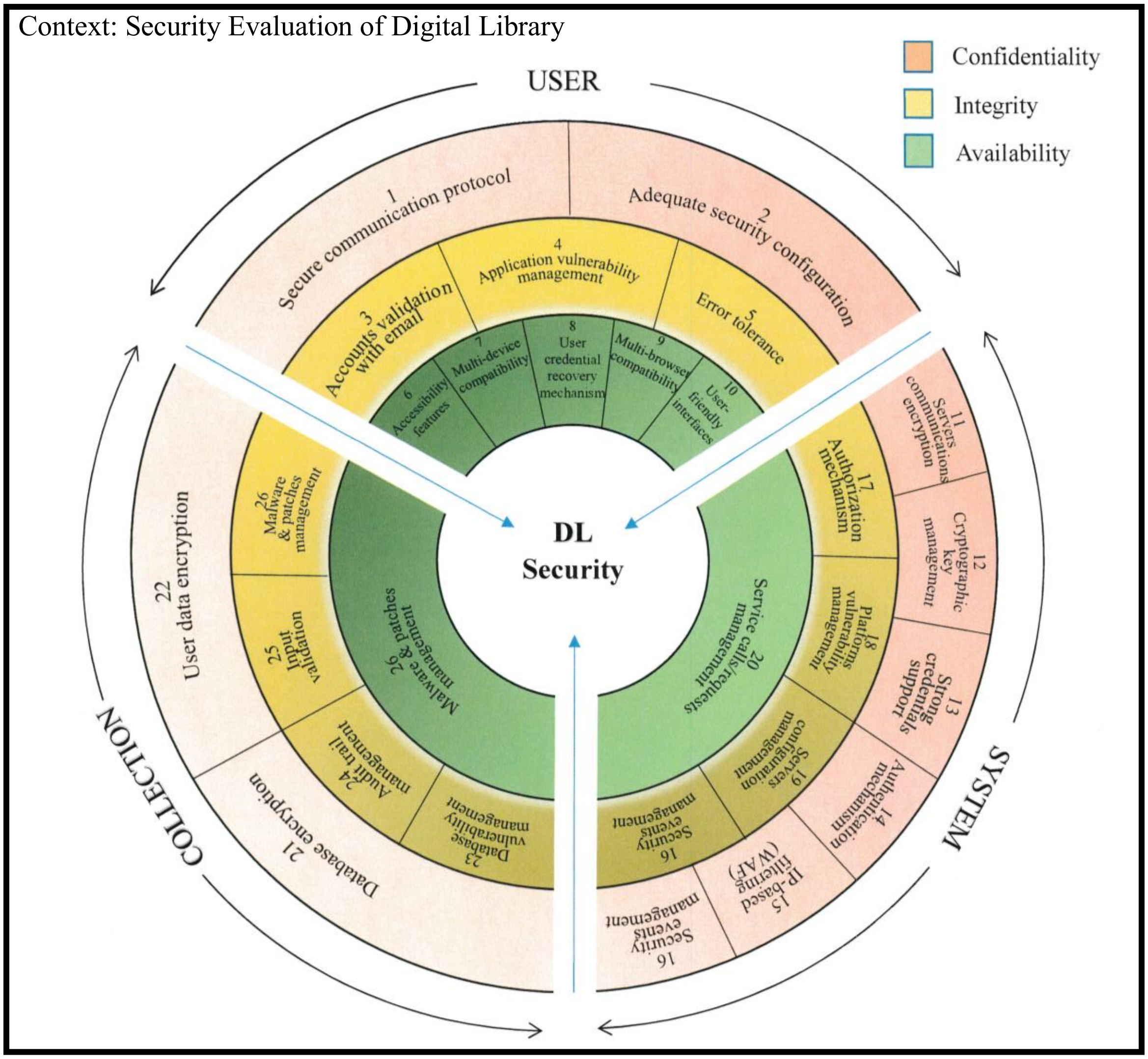

Context: Security Evaluation of Digital Library
Confidentiality
Integrity
Availability
USER
SYSTEM
COLLECTION
1 Secure communication protocol
2 Adequate security configuration
3 Accounts validation with email
4 Application vulnerability management
5 Error tolerance
6 Accessibility features
7 Multi-device compatibility
8 User credential recovery mechanism
9 Multi-browser compatibility
10 User-friendly interfaces
11 Servers communications encryption
12 Cryptographic key management
13 Strong credentials support
14 Authentication mechanism
15 IP-based filtering (WAF)
16 Security events management
17 Authorization mechanism
18 Platforms vulnerability management
19 Servers configuration management
16 Security events management
20 Service calls/requests management
21 Database encryption
22 User data encryption
23 Database vulnerability management
24 Audit trail management
25 Input validation
26 Malware & patches management
26 Malware & patches management
DL Security

Figure 3.1: A model for security evaluation of digital libraries.

The model (Figure 3.1) depicts three DL components (user, system, collection). Each component is divided into three clusters that represent the security attributes of confidentiality (pink), integrity (orange) and availability (green). The idea of having these clusters run through all components as a ring is to demonstrate the need to meet each of these security attributes in each of the DL components for overall security.

Furthermore, each cluster is comprised of several sectors that represent the security items for ensuring the security attribute that the cluster stands for. We further break down the security items in the model into specific security requirements in the checklist (Table 3.2). The checklist is an evaluation tool, which delineates all the specific security requirements an evaluator should check for in a DL for security. The collective fulfilment of the security items established in the model represents the security of digital library. We describe these security items briefly in Table 3.1.

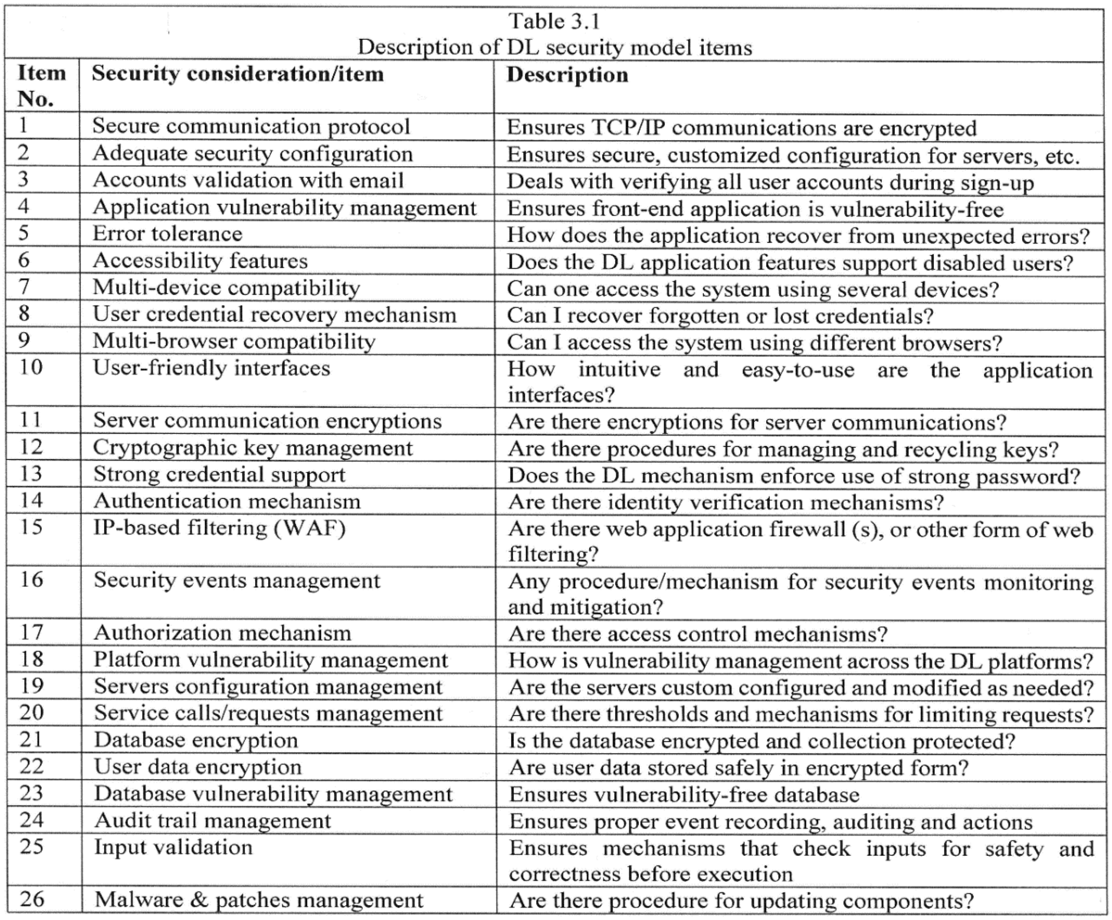


Table 3.1
Description of DL security model items

Item No.	Security consideration/item	Description
1	Secure communication protocol	Ensures TCP/IP communications are encrypted
2	Adequate security configuration	Ensures secure, customized configuration for servers, etc.
3	Accounts validation with email	Deals with verifying all user accounts during sign-up
4	Application vulnerability management	Ensures front-end application is vulnerability-free
5	Error tolerance	How does the application recover from unexpected errors?
6	Accessibility features	Does the DL application features support disabled users?
7	Multi-device compatibility	Can one access the system using several devices?
8	User credential recovery mechanism	Can I recover forgotten or lost credentials?
9	Multi-browser compatibility	Can I access the system using different browsers?
10	User-friendly interfaces	How intuitive and easy-to-use are the application interfaces?
11	Server communication encryptions	Are there encryptions for server communications?
12	Cryptographic key management	Are there procedures for managing and recycling keys?
13	Strong credential support	Does the DL mechanism enforce use of strong password?
14	Authentication mechanism	Are there identity verification mechanisms?
15	IP-based filtering (WAF)	Are there web application firewall (s), or other form of web filtering?
16	Security events management	Any procedure/mechanism for security events monitoring and mitigation?
17	Authorization mechanism	Are there access control mechanisms?
18	Platform vulnerability management	How is vulnerability management across the DL platforms?
19	Servers configuration management	Are the servers custom configured and modified as needed?
20	Service calls/requests management	Are there thresholds and mechanisms for limiting requests?
21	Database encryption	Is the database encrypted and collection protected?
22	User data encryption	Are user data stored safely in encrypted form?
23	Database vulnerability management	Ensures vulnerability-free database
24	Audit trail management	Ensures proper event recording, auditing and actions
25	Input validation	Ensures mechanisms that check inputs for safety and correctness before execution
26	Malware & patches management	Are there procedure for updating components?

3.4 Methodology

We adopt tool-based and qualitative approaches to evaluate CLARK digital library. CLARK is a living repository of cybersecurity curriculum contributed by cybersecurity scholars and professionals from several US institutions. While we use tool-based approach to assess security criterion 3 (vulnerabilities) of the checklist, we use qualitative approach to assess security criteria 1 (Encryption Mechanism), 2 (Identification, Authentication, Authorization Mechanisms), 4 (System/Security Audit) and 5 (Usability & Human-factor Support).

3.4.1 Tool-Based

To investigate vulnerabilities across the digital library components, we use vulnerability scanning/penetration testing tools. Because it is important to choose the right tool(s), we studied several open-source and commercially available tools. Each tool has its unique features and strengths [23]. We identified WebSecurify and Zed Attack Proxy (ZAP) [19] for vulnerability scanning and Burp Suite for penetration testing. We chose WebSecurify and ZAP for vulnerabilities scanning because Websecurify is free, fast, user-friendly and efficient, while ZAP specifically checks for the OWASP top 10 vulnerabilities [19]. ZAP is also multi-operating system compatible, and has better report generation and customization feature. To conduct scan on CLARK DL, we modified the proxy settings on ZAP for the *standard mode* intended for just vulnerability scanning. Burp Suite was selected for penetration testing because of its various scanning and attack features that are very customizable. Its free version offers several uses and comes pre-installed with recent Kali distributions.

3.4.2 Qualitative Approach

Here, assessment is guided by the security checklist we developed (table 3.2). The checklist delineates the specific DL security requirements based on best practices derived from security standards [17,18], common criteria [16], scholarly articles on security metrics [20], etc. We map the security requirements in the checklist to the security items of the model (described in table 3.1) by their item numbers. In the checklist, each section represents a criterion with its set of specific security requirements. The "supports item" column shows how each requirement in the checklist connect to our model's item. "Max" weight is the highest possible score for meeting an associated requirement, while the "earned" weight is the achieved score during evaluation.

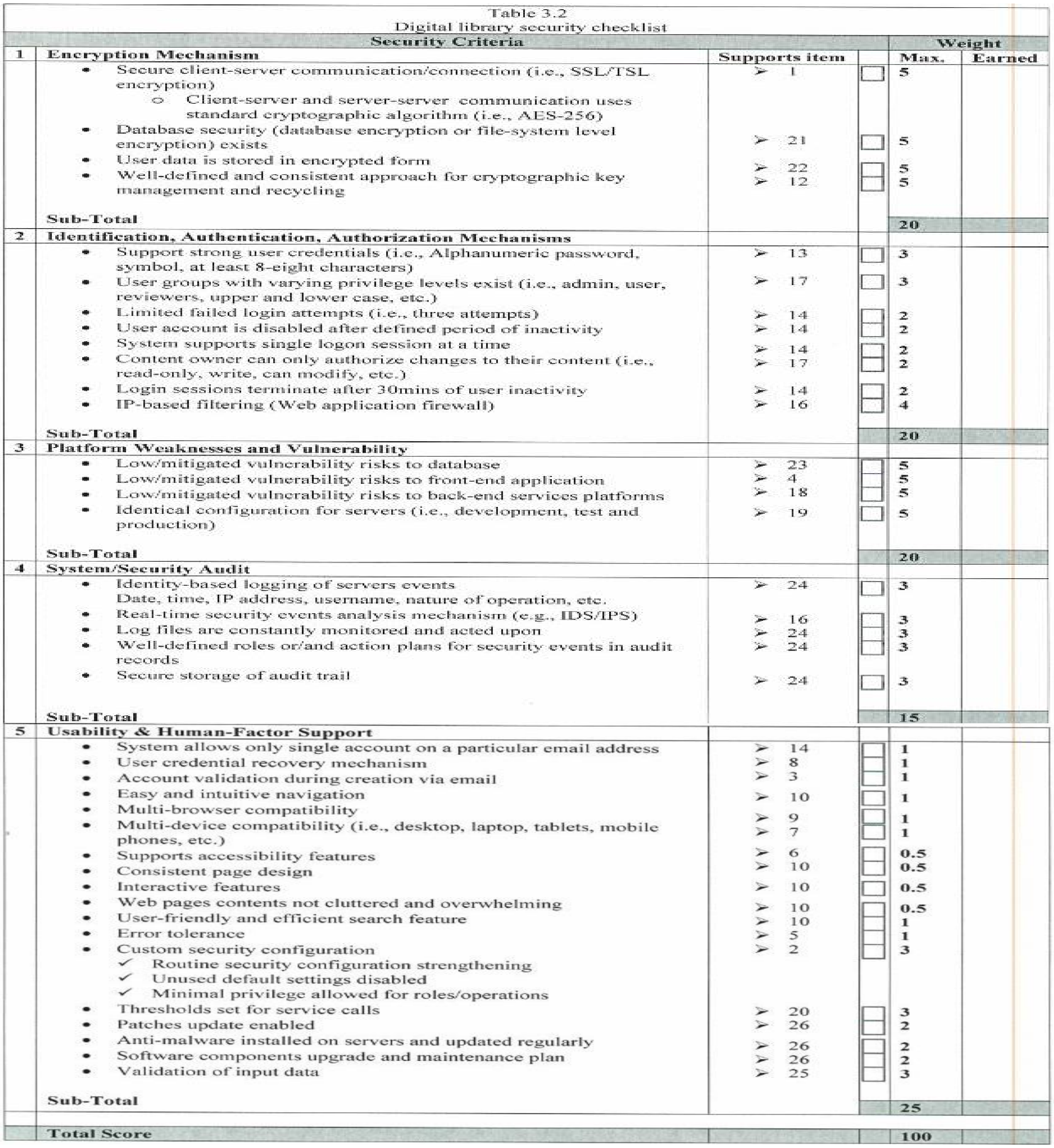


Table 3.2
Digital library security checklist

	Security Criteria			Weight	
		Supports item		Max.	Earned
1	**Encryption Mechanism**				
	• Secure client-server communication/connection (i.e., SSL/TSL encryption)	➢ 1	□	5	
	○ Client-server and server-server communication uses standard cryptographic algorithm (i.e., AES-256)				
	• Database security (database encryption or file-system level encryption) exists	➢ 21	□	5	
	• User data is stored in encrypted form	➢ 22	□	5	
	• Well-defined and consistent approach for cryptographic key management and recycling	➢ 12	□	5	
	Sub-Total			**20**	
2	**Identification, Authentication, Authorization Mechanisms**				
	• Support strong user credentials (i.e., Alphanumeric password, symbol, at least 8-eight characters)	➢ 13	□	3	
	• User groups with varying privilege levels exist (i.e., admin, user, reviewers, upper and lower case, etc.)	➢ 17	□	3	
	• Limited failed login attempts (i.e., three attempts)	➢ 14	□	2	
	• User account is disabled after defined period of inactivity	➢ 14	□	2	
	• System supports single logon session at a time	➢ 14	□	2	
	• Content owner can only authorize changes to their content (i.e., read-only, write, can modify, etc.)	➢ 17	□	2	
	• Login sessions terminate after 30mins of user inactivity	➢ 14	□	2	
	• IP-based filtering (Web application firewall)	➢ 16	□	4	
	Sub-Total			**20**	
3	**Platform Weaknesses and Vulnerability**				
	• Low/mitigated vulnerability risks to database	➢ 23	□	5	
	• Low/mitigated vulnerability risks to front-end application	➢ 4	□	5	
	• Low/mitigated vulnerability risks to back-end services platforms	➢ 18	□	5	
	• Identical configuration for servers (i.e., development, test and production)	➢ 19	□	5	
	Sub-Total			**20**	
4	**System/Security Audit**				
	• Identity-based logging of servers events Date, time, IP address, username, nature of operation, etc.	➢ 24	□	3	
	• Real-time security events analysis mechanism (e.g., IDS/IPS)	➢ 16	□	3	
	• Log files are constantly monitored and acted upon	➢ 24	□	3	
	• Well-defined roles or/and action plans for security events in audit records	➢ 24	□	3	
	• Secure storage of audit trail	➢ 24	□	3	
	Sub-Total			**15**	
5	**Usability & Human-Factor Support**				
	• System allows only single account on a particular email address	➢ 14	□	1	
	• User credential recovery mechanism	➢ 8	□	1	
	• Account validation during creation via email	➢ 3	□	1	
	• Easy and intuitive navigation	➢ 10	□	1	
	• Multi-browser compatibility	➢ 9	□	1	
	• Multi-device compatibility (i.e., desktop, laptop, tablets, mobile phones, etc.)	➢ 7	□	1	
	• Supports accessibility features	➢ 6	□	0.5	
	• Consistent page design	➢ 10	□	0.5	
	• Interactive features	➢ 10	□	0.5	
	• Web pages contents not cluttered and overwhelming	➢ 10	□	0.5	
	• User-friendly and efficient search feature	➢ 10	□	1	
	• Error tolerance	➢ 5	□	1	
	• Custom security configuration ✓ Routine security configuration strengthening ✓ Unused default settings disabled ✓ Minimal privilege allowed for roles/operations	➢ 2	□	3	
	• Thresholds set for service calls	➢ 20	□	3	
	• Patches update enabled	➢ 26	□	2	
	• Anti-malware installed on servers and updated regularly	➢ 26	□	2	
	• Software components upgrade and maintenance plan	➢ 26	□	2	
	• Validation of input data	➢ 25	□	3	
	Sub-Total			**25**	
	Total Score			**100**	

We adapt the mini-Delphi technique to assign weights and scores to the security criteria and requirements respectively. The scores and weights were assigned by the team of experts, comprised of programmers, usability experts, system/software architects, and information system specialists. Ranking was based on their perceived unique impact of each criterion on the security of a DL (or any IS) through rounds of scoring, justification and reconciliation. The varying weightings of the criteria sum up to a total of 100 achievable points. Eventually, the checklist was harmonized to capture all relevant security requirements, testable through vulnerability scanning, expert review of functionalities, and usability testing.

To use the checklist to evaluate CLARK, we investigated and checked off all boxes for the security requirements met by CLARK. Next, we summed up all earned points within each criterion to get the subtotals for each. Finally, we were able to arrive at the total score achieved by the system by summing up all the subtotals. The following section presents the results of this study.

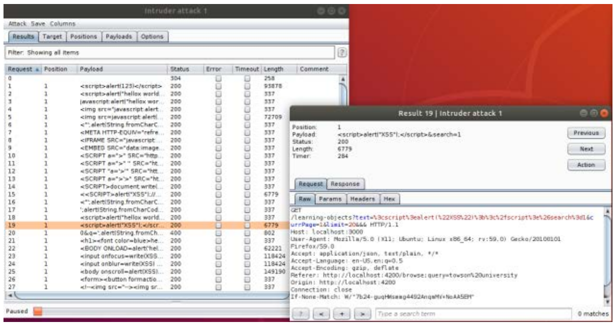
4.0 RESULTS

In this section, we present results for both the tool-based and qualitative assessment.

4.1 Tool-Based

Vulnerability scanning on CLARK using WebSecurify shows that the cybersecurity DL passed 80% of the OWASP top ten security risk assessment giving CLARK an A grade. In addition, the vulnerability scanning performed using ZAP tool resulted in two false positive alerts: A low-risk, third-party domain script detection and a path traversal error marked as high risk. While the first error is due to the discrepancy in hostnames of CLARK's website and that of the external script (i.e., google analytics, which is safe), ZAP threw the second flag, having found "**etc**" in the word "F**etc**hes" that it literally parsed. This similar flag is common whenever ZAP detects strings like "bin" or "boot", and so on. In this case, both alerts pose no security risks to the system.

Modifying proxy settings (amongst other configuring) on Burp Suite, and with the use of Burps fuzzer tool (capable of identifying injection, buffer overflow and cross-site scripting (XSS)), we further conducted penetration testing on CLARK. With Burps' intruder tool, and using some common XSS attacks retrieved online and uploaded into the payload options, results for the inserted payloads is as shown in figure 4.1.



Request	Position	Payload	Status	Error	Timeout	Length	Comment
0			304	□	□	258	
1	1	<script>alert(123)</script>	200	□	□	93878	
2	1	<script>alert("hellox world…	200	□	□	337	
3	1	javascript:alert("hellox wor…	200	□	□	337	
4	1	<img src="javascript:alert…	200	□	□	337	
5	1	<img src=javascript:alert(…	200	□	□	72709	
6	1	<"';alert(String.fromCharC…	200	□	□	337	
7	1	<META HTTP-EQUIV="refre…	200	□	□	337	
8	1	<IFRAME SRC="javascript:…	200	□	□	337	
9	1	<EMBED SRC="data:image…	200	□	□	337	
10	1	<SCRIPT a=">" SRC="http…	200	□	□	337	
11	1	<SCRIPT a=">" '' SRC="ht…	200	□	□	337	
12	1	<SCRIPT "a='>'" SRC="htt…	200	□	□	337	
13	1	<SCRIPT a=">'>" SRC="ht…	200	□	□	337	
14	1	<SCRIPT>document.write(…	200	□	□	337	
15	1	<<SCRIPT>alert("XSS");//…	200	□	□	6779	
16	1	<"';alert(String.fromCharC…	200	□	□	337	
17	1	';alert(String.fromCharCod…	200	□	□	337	
18	1	<script>alert("hellox world…	200	□	□	337	
19	1	<script>alert("XSS");</scr…	200	□	□	6779	
20	1	0&q=';alert(String.fromCh…	400	□	□	802	
21	1	<h1><font color=blue>he…	200	□	□	337	
22	1	<BODY ONLOAD=alert('hel…	200	□	□	62221	
23	1	<input onfocus=write(XSS…	200	□	□	118424	
24	1	<input onblur=write(XSS) …	200	□	□	118424	
25	1	<body onscroll=alert(XSS)…	200	□	□	149190	
26	1	<form><button formactio…	200	□	□	337	
27	1	<!--<img src="--><img sr…	200	□	□	337	

Position: 1
Payload: <script>alert("XSS");</script>&search=1
Status: 200
Length: 6779
Timer: 284

```
GET
/learning-objects?text=%3cscript%3ealert(%22XSS%22)%3b%3c%2fscript%3e%26search%3d1&c
urrPage=1&limit=20&&& HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:59.0) Gecko/20100101
Firefox/59.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:4200/browse;query=towson%20university
Origin: http://localhost:4200
Connection: close
If-None-Match: W/"7b24-guqHMsemg4492AnqeMY+NoAA5EM"
```

Figure 4.1: Results from Burp Suite intrusion on CLARK

With the first result being the baseline request, it is important to point out that the closer the length of a request is to the length of the baseline, the more likely the payload was not harmful to the application. Here, we see that majority of the payloads gave a 200 response, which means that the status is okay.

4.2 Qualitative Study

The results of the qualitative study on CLARK are presented based on the criteria and specific security requirements defined in our checklist (Table 3.2).

Encryption Mechanism

In this criterion, which examines the overall encryption mechanisms, evaluation results indicate that CLARK has secure transport layer encryption using TLS/SSL (Transport Layer Security/Secure Sockets Layer). This fulfills client-server and server-server communication security requirement. CLARK database runs on MongoDb, which runs on Amazon Web Services (AWS) engine, and uses 256-bit Advanced Encryption Standard in Galois/Counter Mode (AES256-GCM) for the Linux operating system, and AES256-CBC for the Windows OS [21]. In addition, user data are hashed while cryptographic key are encrypted and stored with strong cipher as defined by AWS. With both the database and transport layer communications across services secure, CLARK certifies this criterion, scoring the full 20 points for this category.

Authentication and Authorization Mechanism: Although CLARK shows limited or no controls for some of the security requirements under this criterion, such as support for strong password, limited failed login, single sign-on support, inactive session termination, it meets some key requirements which includes authorization mechanism and web application firewalls (WAFs), for the services and the database. Overall, CLARK performed relatively low under this criterion as seen in Table 4.1.

Table 4.1
Consolidated Evaluation Results

	Score	
Security Criteria	**Achievable**	**Earned**
Encryption Mechanism	20	20
Authentication and Authorization Mechanism	20	8
Platform Weaknesses and Vulnerability	20	18
System/Security Audit	15	12
Usability & Human-Factor Support	25	20.5
Total	**100**	**78.5**

Platform Weakness and Vulnerability: The quantitative assessment results of CLARK as presented in 4.1 shows CLARK passed 80% of the OWASP's top ten security risks. CLARK's database is considered secure, enjoying all the standard protection/shield offered by AWS that it runs on. The system also uses Docker to encapsulate all tools in identical containers for the development, test and production environment, thereby ensuring configuration consistency for those environments that the system traverse during its development cycle. All these are geared towards minimizing vulnerabilities, affording CLARK a score of 18 out of 20 for this criterion.

System/Security Audit: Our investigation shows that CLARK captures all events using Amazon CloudWatch, a monitoring service for AWS cloud resources and applications, while it uses Sentry to track application errors, log and report to admins for resolution. These two features satisfy this requirement with a 12 out of 15 points earned.

Usability & Human-Factor Support: Although accessibility features were yet to be adequately implemented, CLARK has an overall good outlook when it comes to usability of its web application.

Additionally, we found that error tolerance is another lagging area as some failures – if they occur – may result in a blunt crashed interface. We also found that AWS Identity and Access Management (IAM) *super user* can and does grant minimal privilege as he creates other users – a strategy for checking inappropriate use and reducing human-factor threats. Greenkeeper handles application libraries (and other dependencies, security patches, etc.) updates, while all backend infrastructures/services that run on AWS are being maintained and updated by AWS. These would ensure normal system-user interactions for reducing security incidents. With a score of 20.5 out of 25, CLARK boasts of quality user experience and adequate support for user-system interactions.

5. DISCUSSION & CONCLUSIONS

In this study, we identified the key security requirements of a digital library through extensive review of literature on evaluations studies, security standards, security reports, and so on. Considering these requirements, we developed a model as our contribution and a tool (checklist) for guiding system developers, evaluators, and system administrators on the requirements for ensuring security of digital libraries. We also evaluated CLARK cybersecurity digital library, as to test the effectiveness of the model, adopting tool-based and qualitative assessment approach.

For the tool-based study, we used free, ease-to-use, speedy and automatic testing tools that offer valuable penetration testing phases under a single framework; however, their limitations are often in the report of false positives that require efforts to confirm that the alerts are not harmful. In both the vulnerability scanning (with WebSecurify and ZAP) and penetration testing (with Burp suite), CLARK did well with an eighty percent score. As for the qualitative evaluation of CLARK, summing up the scores for all the criteria resulted in an overall score of 78% out of 100%, implying that CLARK is considerably secure. The successful use of our model and checklist to evaluate CLARK demonstrates the effectiveness of our model for evaluating any other digital library. We plan to further investigate that, in our next study, by evaluating multiple digital libraries.

REFERENCES

[1] Carlo Bertot, John, et al. "Functionality, usability, and accessibility: Iterative user-centered evaluation strategies for digital libraries." *Performance Measurement and Metrics* 7.1 (2006): 17-28.

[2] Tsakonas, Giannis, Sarantos Kapidakis, and Christos Papatheodorou. "Evaluation of user interaction in digital libraries." *Notes of the DELOS WP7 workshop on the evaluation of Digital Libraries, Padua, Italy*. 2004.

[3] Blandford, Ann, et al. "The PRET A Rapporter framework: Evaluating digital libraries from the perspective of information work." *Information Processing & Management* 44.1 (2008): 4-21.

[4] Nicholson, Scott. "A conceptual framework for the holistic measurement and cumulative evaluation of library services." *Proceedings of the American Society for Information Science and Technology* 41.1 (2004): 496-506.

[5] Borgman, Christine L., et al. "Evaluating digital libraries for teaching and learning in undergraduate education: a case study of the Alexandria Digital Earth Prototype (ADEPT)." (2000).

[6] Gonçalves, Marcos André, et al. "Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries." *ACM transactions on information systems (TOIS)* 22.2 (2004): 270-312.

[7] Cybersecurity Labs and Digital Knowledgebase (CLARK) digital library. Effective cybersecurity curriculum at your fingertips, clark.center/home. Accessed 27 March 2019.

[8] Saracevic, Tefko. "Digital library evaluation: Toward an evolution of concepts." (2000).

[9] Jeng, Judy. "Usability assessment of academic digital libraries: effectiveness, efficiency, satisfaction, and learnability." *Libri*55.2-3 (2005): 96-121.

[10] Hoe-Lian Goh, Dion, et al. "A checklist for evaluating open source digital library software." *Online Information Review*30.4 (2006): 360-379.

[11] Wang, Ju An, et al. "Security metrics for software systems." *Proceedings of the 47th Annual Southeast Regional Conference*. ACM, 2009.

[12] Maconachy, W. Victor, et al. "A model for information assurance: An integrated approach." *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*. Vol. 310. United States Military Academy, West Point. IEEE, 2001.

[13] Fuhr, Norbert, et al. "Evaluation of digital libraries." *International Journal on Digital Libraries* 8.1 (2007): 21-38.

[14] Conklin, William Arthur, Dan Shoemaker, and Anne Kohnke. "Cyber resilience: Rethinking cybersecurity strategy to build a cyber-resilient architecture." *ICMLG2017 5th International Conference on Management Leadership and Governance*. 2017.

[15] Alexander, Eldridge "Part 1: Usability is Security." *Duo Labs*, duo.com/blog/part-1-usability-is-security. Accessed 27 march 2018.

[16] *The Common Criteria*. Publications, www.commoncriteriaportal.org/cc/. Accessed 27 March 2019.

[17] *National Institute of Standards and Technology.* Computer and Security Resource Center, csrc.nist.gov/projects/cryptographic-module-validation-program/standards. Accessed 27 March 2019.

[18] *International Organization for Standardization.* ISO 27001. www.iso.org/isoiec-27001-information-security.html. Accessed 27 March 2019.

[19] *The OWASP Foundation*. www.owasp.org/index.php/Main_Page, Accessed 27 March 2019.

[20] Swanson, Marianne M., et al. *Security metrics guide for information technology systems*. No. Special Publication (NIST SP)-800-55. 2003.

[21] Kaza, Siddharth, Blair Taylor, and Kyle Sherbert. "Hello, World!—Code Responsibly." *IEEE Security & Privacy* 16.1 (2018): 98-100.

[22] *TechBeacon,* 9 code and framework trends to watch in 2018, https://techbeacon.com/app-dev-testing/9-code-framework-trends-watch-2018. Accessed 27 March 2019.

[23] Fonseca, Jose, Marco Vieira, and Henrique Madeira. "Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks." *13th Pacific Rim international symposium on dependable computing (PRDC 2007)*. IEEE, 2007.