Nigerian Research Journal of Engineering and Environmental Sciences 4(2) 2019 pp. 724-739

Nigerian Research Journal of Engineering and Environmental Sciences
Journal homepage: www.rjees.com

Original Research Article

Adoption of Internet Voting Platform Containing Data Injection Threats with Structured LINQ

***[1]Nwankwo, W. and [2]Njoku, C.C.**
[1]Department of Computer Science, Edo University Iyamho, Edo State, Nigeria.
[2]Department of Computer Science Education, Aminu Saleh College of Education, Azare, Bauchi State, Nigeria.
*nwankwo.wilson@edouniversity.edu.ng

ARTICLE INFORMATION

Article history:
Received 08 Oct, 2019
Revised 18 Nov, 2019
Accepted 24 Nov, 2019
Available online 30 Dec, 2019

Keywords:
Internet voting
LINQ
Stored procedures
Parameterized queries
Data injection
Vulnerability

ABSTRACT

Web-based and other electronic voting platforms are growing as some countries have considered them viable in ensuring transparency and accountability during elections. One of the most cited challenges that hinders widespread adoption borders on cybersecurity. In modern times, attackers and cyber criminals have developed automated tools that could exploit seemingly secured data intensive applications within few minutes resulting to data breaches and losses. Data injection attacks (DIA) are prevalent on web applications and software engineers are combatting the trend using techniques such as parameterized queries (PQ), stored procedures (SP), and language integrated query (LINQ). This paper is aimed at evaluating the effectiveness of LINQ in circumventing DIA directed to internet voting application (IVP). This paper employed a hybrid approach comprising object-oriented techniques (OOT), exploitation, and vulnerability analysis respectively. OOT was used to develop two streams of IVP with C# as the base language. The two software streams were: Fully LINQ-based, and embedded SQL-based with LINQ. Vulnerability analysis were conducted on both application streams using the SQLMAP installed on kali Linux. The results showed that a LINQ-only platform offered good resistance to data injection, whereas the application stream with traditional embedded SQL was susceptible to exploitation with SQLMAP. It is concluded that the use of carefully structured LINQ could effectively circumvent web-based injection attacks in IVPs.

1. INTRODUCTION

Electronic voting platforms use information communication technologies to enhance pre-election (voter registration, voter accreditation), election (vote casting), and post-election activities (election results processing, publishing, etc.). Though there are various perspectives on what constitutes an electronic voting platform (de Vuyst and Fairchild, 2005; Ofori-Dwumfuo and Paatey, 2011; Achieng and Ruhode, 2013; Idike, 2014; Okoro, 2016; Uzedhe and Okhaifoh, 2016; Alausa and Akingbade,2017) the context in which

Rouse (2011) describes it as one that "allows a voter to record his or her secure and secret ballot electronically" is generic.

Since 2009, when Nigeria returned to democratic rule, the traditional ballot-based system of election has been in use. Suffice it to say that we are yet to witness transparency during the elections (Mediayanose, 2018; Uwujiaeren et al., 2019; Odunsi, 2019; Reuters, 2019). The story is not different in other parts of Africa. Developing countries are the most affected in the trend of election challenges (Onimisi, 2011; Aranuwa and Oriola, 2012; Ahmad et al, 2015; Nwogu, 2015)

Elections in any democratic government may be conducted through one or a mix of two popular platforms: tradition ballot-based system, and the electronic voting platforms. The former is the most popular and has been in used for centuries. Owing to the weaknesses (not to be attributed wholly to the process itself but the executors and stakeholders in the system) there has been a rising call for adoption of electronic voting platforms across Africa. According to Laukkonen (2018), three main variants of electronic voting exists and this includes: optical scanning, internet voting(i-voting), and direct recording. Optical scanning is recognized as the oldest electronic voting option followed by direct recording. Direct recording seems to be the commonest in modern times whereas i-voting is gaining some momentum. Electronic voting systems are being used in developed countries such as Germany, Canada, India, Brazil, Norway, Switzerland etc. A good electronic voting platform should enable all legitimate voters to participate in elections. Validation of voters should be by way of any government approved means of identification like Driver's Licence, National Identity Card, and International Passport or electronic means like biometric capture (Alausa and Akingbade, 2017).

Venezuela, India, Brazil, and Estonia are countries who use electronic voting nationwide. A remarkable observation is that no single country uses only one form of electronic voting but a mix of available options. Majority of those countries who has adopted it combine it with the traditional ballot-based system.

Notwithstanding its advantages, electronic voting systems have peculiar challenges such as susceptibility to manipulation by hackers (Al-Ameen and Talab, 2013; Javaid, 2014), malwares on compromised connected devices, unstable power supply (especially in developing countries), hardware and software failure, etc. In Nigeria, the use of electronic voting is practicable but the contending factors especially instrumentation and cyber security (Nwankwo and Ukaoha, 2019; Nwankwo and Olayinka, 2019) must be addressed appropriately prior to implementation. Thus, this paper is aimed at addressing the data injection threats to which internet voting applications may be subjected to.

2. MATERIALS AND METHODS

2.1. Hardware

The following pieces of hardware were used in this study:

i. PC@3.5Ghz 16GB RAM, 500GB Hard disk running Ubuntu 18.04 OS

ii. HP Proliant DL-380 G7 Series Server Quad-core Intel Xeon processor@3.8Ghz 32GB RAM running Ubuntu 18.04 Server

iii. HP elitebook 820 (Intel core i7@2.6Ghz, 16GB RAM, 1TB Hard disk) with notebook running Kali Linux distribution.

2.2. Software

Two categories of software were used in this study: Application design and implementation software, and Penetration testing software. The following design and implementation software were utilized:

i. Microsoft Visual studio 2019 community edition

ii. Microsoft SQL Server 2012 Database Management System

The penetration testing software used are:

i. Windows subsystem for Linux 2.0

ii. Kali Linux with the following installed: Metasploit framework 5.0.5 with Armitage, Nmap and Msfconsole, Websploit, SQLMap

2.3. Design of Experiment

A hybrid approach comprising object-oriented techniques (OOT), exploitation, and vulnerability analysis respectively. OOT was used to develop two streams of web-based electronic voting solutions with C# as the base language. The two software streams were: Fully LINQ-based, and embedded SQL-based with LINQ. The two web applications were subjected to white box and black box exploitation and vulnerability analysis using websploit, SQLMAP respectively.

2.4. Application Component Analysis

Figure 1 shows the component model of the internet voting platform. The model is intended to support voter registration, documentation, verification, voter management, vote casting, results processing, and intelligence reporting.

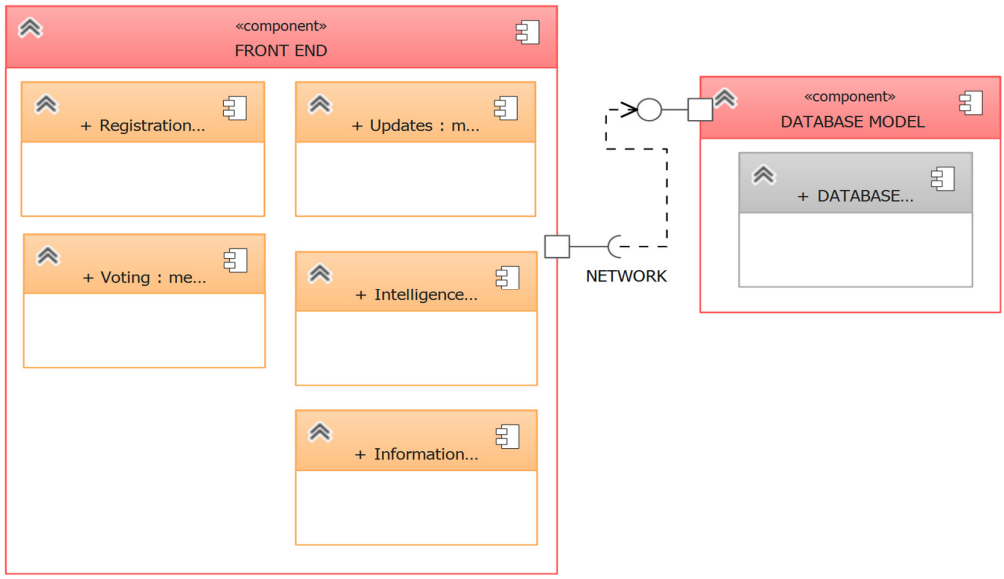

Figure 1: Component model of the internet voting platform

The model is designed to be flexible, so it could be extended to support future needs. All the transactions would be routed to the central database. The subsystems in the system were grouped into two (2) components:

i. The database (backend)

ii. The application program (frontend)

2.5. Modeling Interaction in the Internet Voting System

This is done using a sequence diagram (SD). Typically, the SD depicts interactions among objects over a certain period (Nwankwo, 2015). A sequence diagram shows the following:

i. The external actors including hackers

ii. The messages or methods that are invoked by these actors

iii. The return values (if any) associated with previous messages

iv. The areas or points where loops or iterations exist.

Figure 2-4 show the sequence diagram for the different components of the internet voting platform. In Figure 2, the voter is first registered by the registration officer. Registration succeeds only if the voter data captured is valid and consistent with the system's requirements. After the registration, an identification code or password is given to the voter which enables he/she (voter) to log in. Also, the voter's registration is authenticated - a phase to verify that the voters have access rights and franchise. The output is either request granted or denied. The voter's data are sent directly to the database via a communication network.

2.6. Logical Data Design

The logical model defines the entities that would be rendered by the application program, and what policies and rules would be used to operate on those entities. It is made up of two relatively sub-models, the logical data model and the logical object model. The logical data model is responsible for documenting the properties of entities/objects in the system whereas the logical object model contains the rules/algorithms that operate on the entities, how these rules are grouped into interfaces and classes, and how the various objects interact among themselves to solve macro level requirements. Object-oriented design does not only consider the data attributes of an object but the behaviour of those objects in relation to other objects in the domain. Behaviour may be internal within a given object or external (may be brought about by another object in the domain). These behaviours must be identified and modelled accordingly to enable object-oriented program development. Object behaviour is embedded in a method. Figure 5 shows the logical object model of the system.

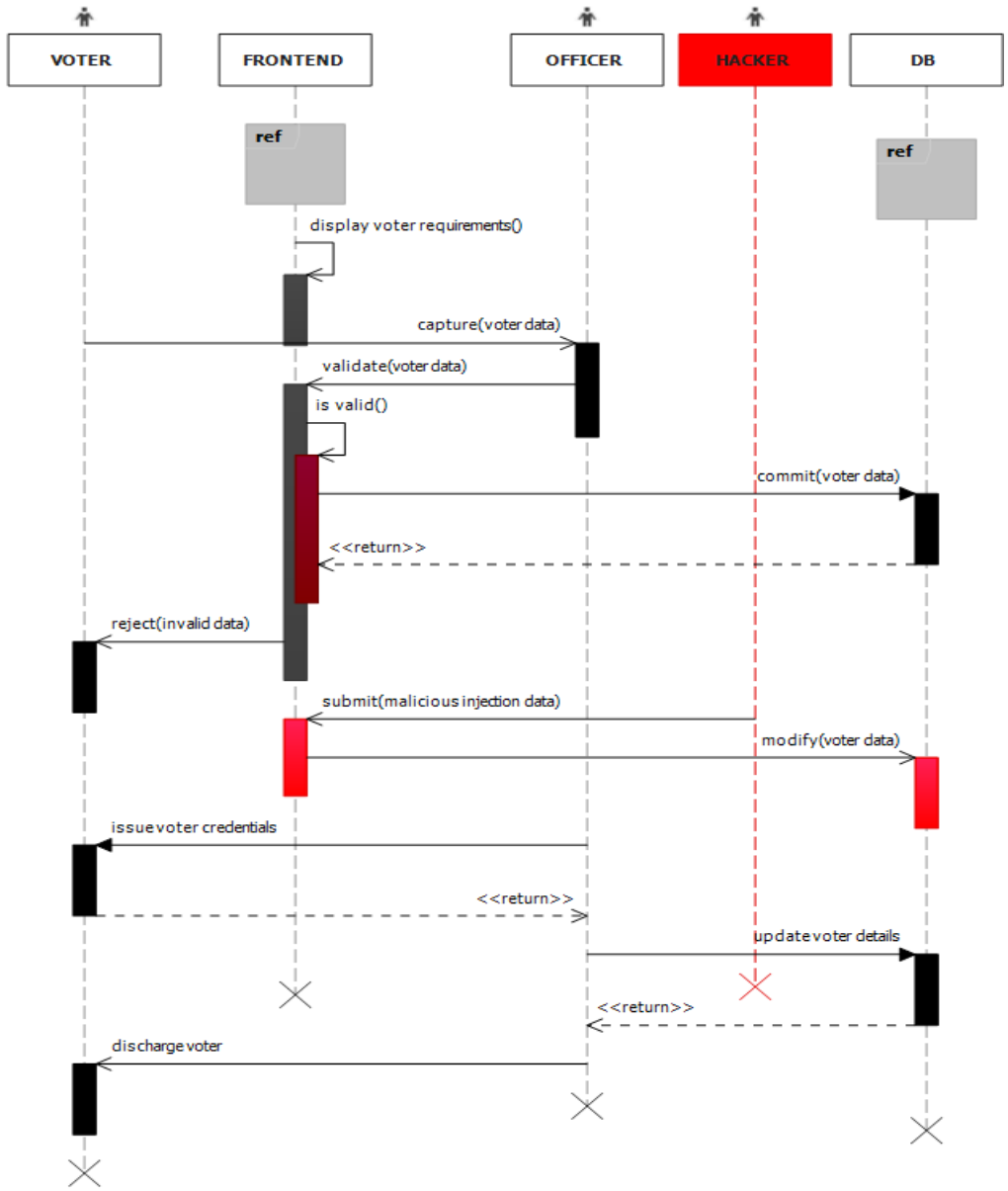

Figure 2: SD for voter registration

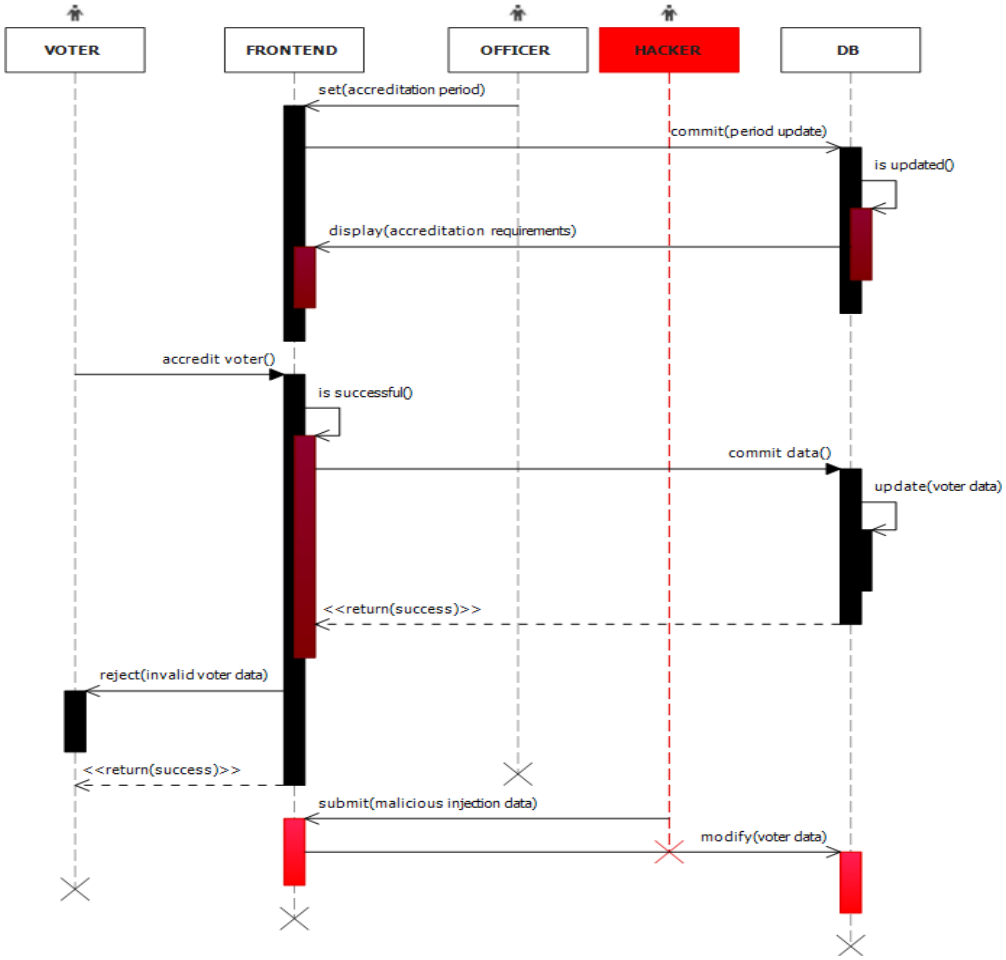

Figure 3: SD for accreditation of voters

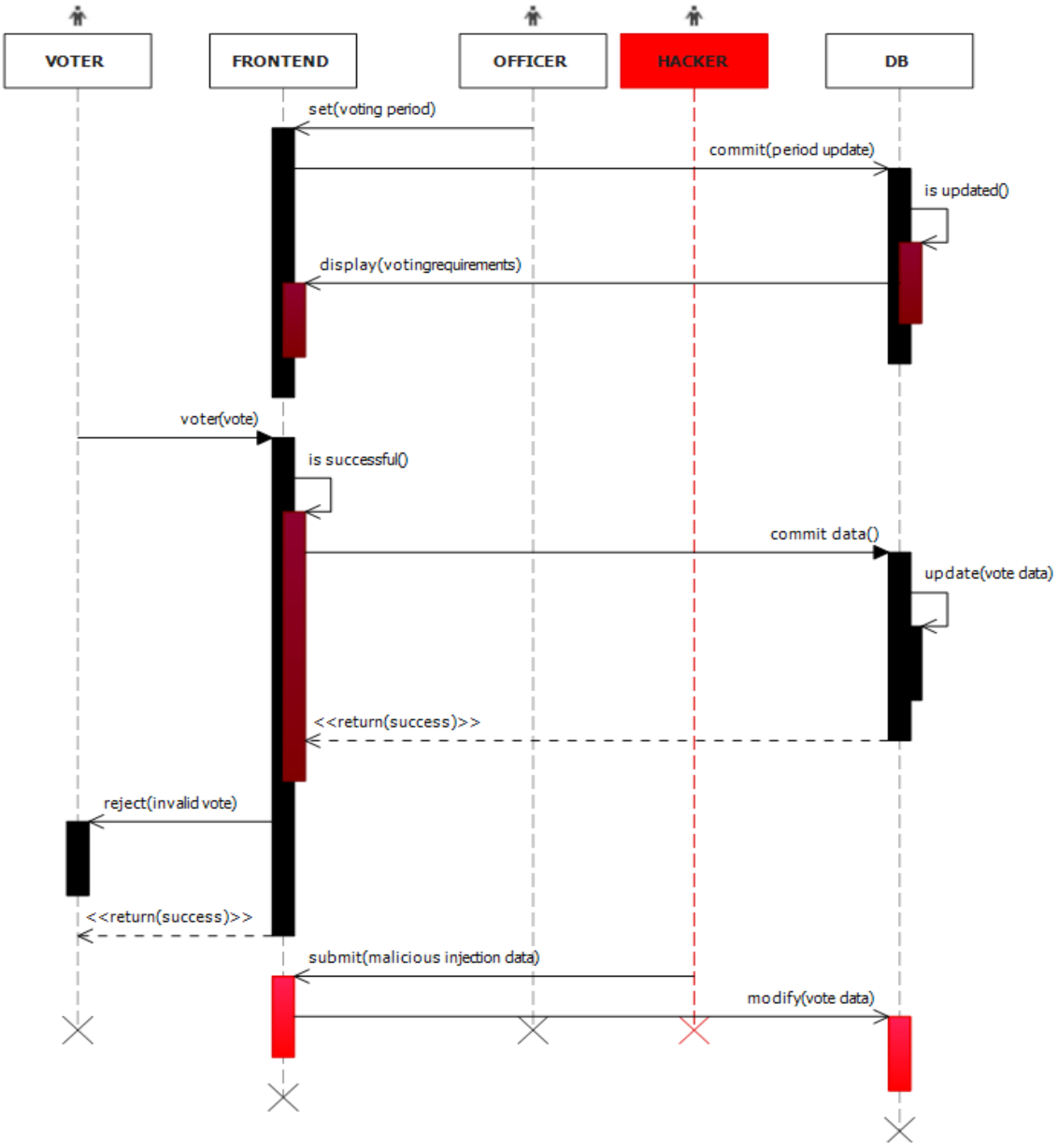

Figure 4: SD for the voting process

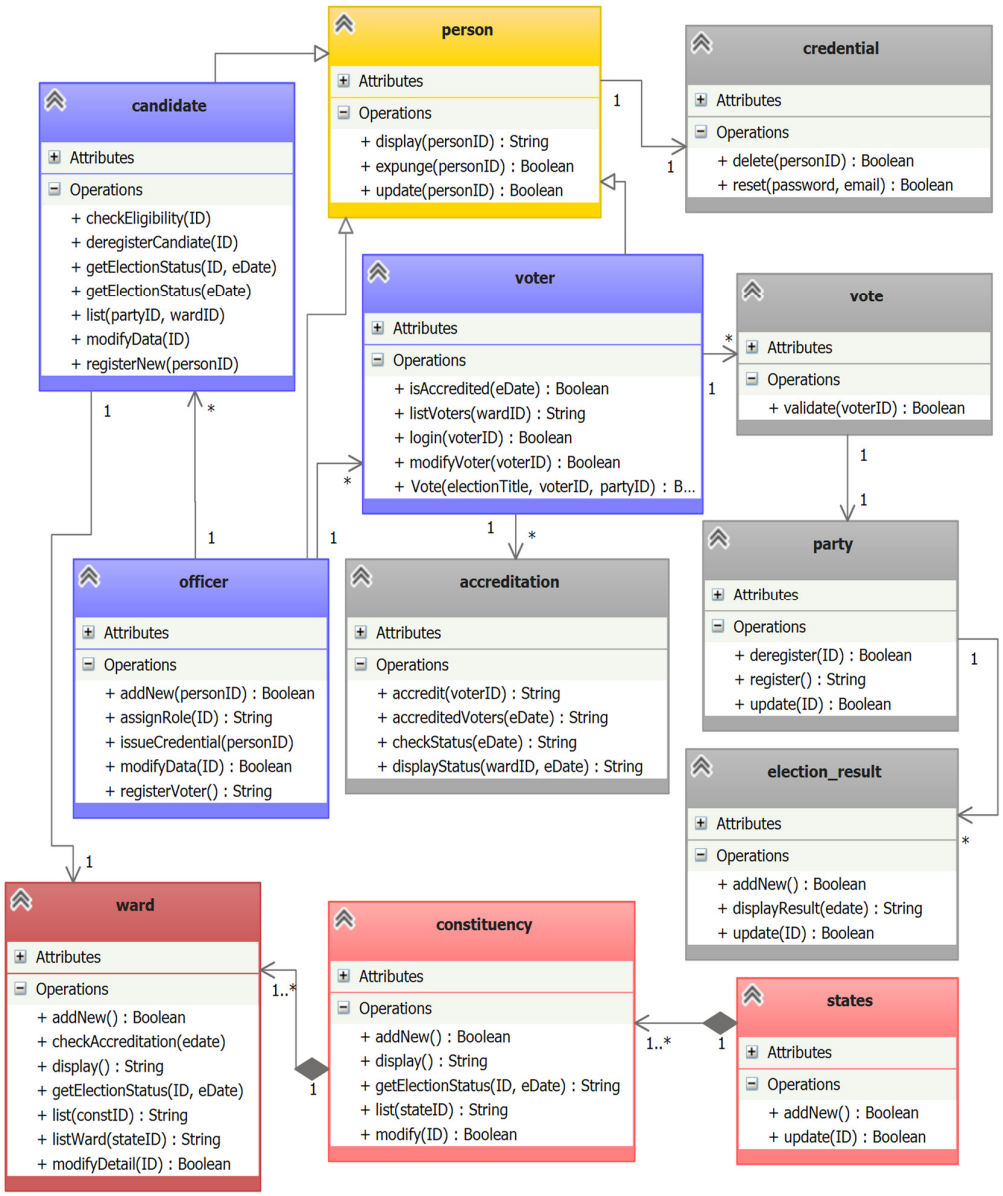

Figure 5: Logical object model showing behaviour of objects in the hierarchy

2.7. Approach to Functional and Penetration Testing

To test the functionality of the implemented i-voting system, structured test types and test cases were used. Some of the test cases are presented in Tables 1-3. Similarly, to test the comparative vulnerabilities of the implemented systems, the inject attack cycle as presented in Figure 6 was used. The injection attack cycle represents the procedure used by majority of data injection attackers to exploit vulnerability of websites as well as carry out intended attacks. In this approach, careful specific blackbox test cases were used to examine the response to values introduced to compromise both application streams. Some of the test cases are presented in Tables 4-5.

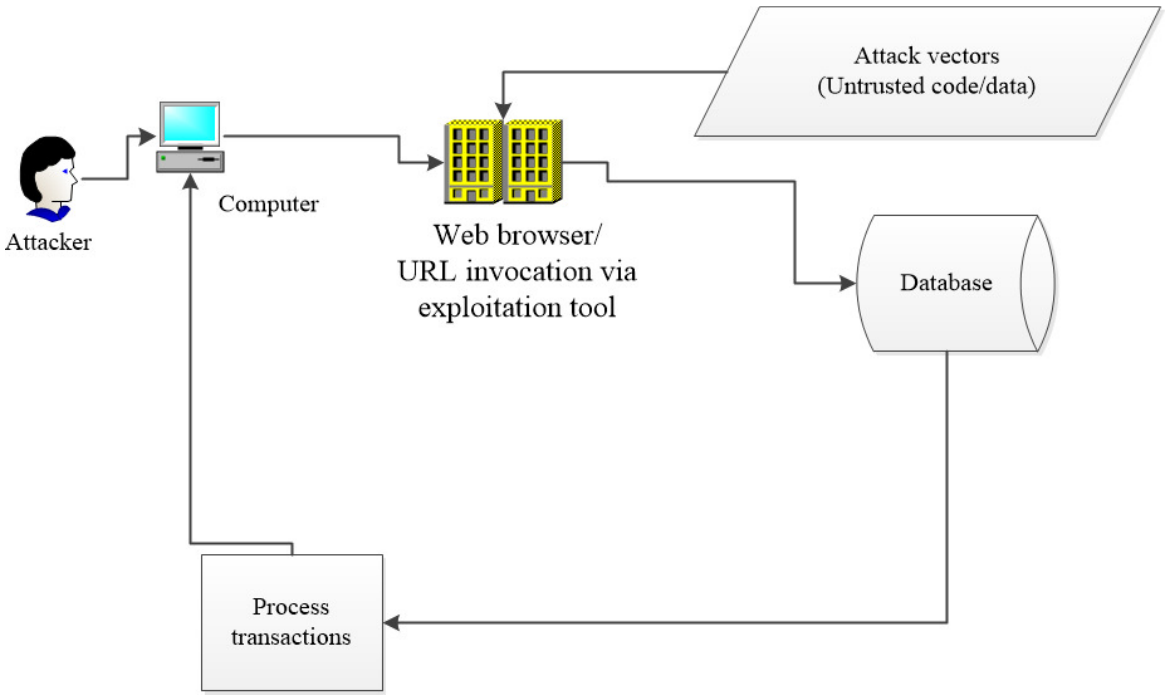

Figure 6: Injection attack cycle

Table 1: Voter registration test case

Test Case ID	VRTC
Unit to test	Register Voter Module
Role	Commission_Officer
Assumptions	All requisite voter's data are captured at this point and stored in the voter registration table of the voter database
Test data	voter bio data, contact details, registration details, qualification and requisite documents
Flow	Accredited INEC officer captures voter data from completed forms and associated documents
Expected result	Captured data is verified, validated and saved in the database
Actual result	Voter registration succeeded
Pass/Fail	Pass

Table 2: Voter accreditation test case

Test Case ID	VATC
Unit to test	Accreditation voter module
Role	System
Assumptions	All registered voters who present themselves on the day of a given election are duly accredited and stored in the accredited voters table in the election voter
Test data	Voter ID, fingerprint, photo
Flow	System uses voter authentication data to validate and accredit voter
Expected result	Captured data is verified, validated and saved in the database
Actual result	voter accreditation succeeded
Pass/Fail	Pass

Table 3: Vote casting test case

Test Case ID	VCTC
Unit to test:	Voting Module
Role	System
Assumptions	All duly accredited voters on the IVP on the day of a given election have access to a carefully controlled Network-enabled computer through which the i-voting application is made available to the voter; voter is presented with accredited candidates of the various parties; voter casts his/her vote; and vote is stored in the
Test data	Voter ID, Parties, Candidates for election
Flow	Voter casts his/her vote against a candidate for a particular category; vote casted is validated and verified against eligible voters, election timeline, and candidates for election respectively
Expected result	Captured vote casted is verified, validated and saved in the database
Actual result	Vote casting succeeded
Pass/Fail	Pass

Table 4: Voter login exploitation test case

Test Case ID	VLETC
Rationale	Vulnerability assessment of Different IVPs
Role	Voter
Assumptions	Blackbox test of susceptibility of Login module to SQL Injection
Test data	Fake Voter ID, fake password
Flow	Penetration tester tries to gain access to the database by issuing a false voter ID and password
Expected result	Test reveals backend server model and where error occurred on Non-LINQ application but conceals such information on a LINQ-based system
Actual result:	Relative
Pass/Fail:	Pass

Table 5: Voter register exploitation test case

Test Case ID	VRETC
Rationale	Vulnerability assessment of Different IVPs
Role	COMMISSION_OFFICER
Assumptions	Blackbox test of susceptibility of voter database display to SQL Injection
Test data	Fake voter table parameters
Flow	Penetration tester tries to gain access to display the voter database by issuing false parameters
Expected result	Test reveals database properties such tables, columns, etc on Non-LINQ application but terminates on a LINQ-based system
Actual result:	Relative
Pass/Fail:	Pass

3. RESULTS AND DISCUSSION

3.1. User Interfaces on the Internet Voting Platform

Like every web application tailored to be accessed by legitimate users, the i-voting platform for the two streams-LINQ-based and non-LINQ-based have features to enable access to only registered voters. Figure 7-8 show the administrative dashboard and voter dashboard respectively. Under the administrative dashboard, there is the 'General' menu that contains the following sub-menus: Elections, Accreditation settings, Candidates and Voters. Through the 'Registered Parties' menu, the names of all registered parties and their logos could be accessed, and more parties could also be registered. The 'Ward' menu houses all the wards used in the system. The 'Candidates' menu shows the qualified candidates and accredited candidates. The 'Registration' menu provides the platform for voter registration, view registered voters, and also register candidates. The 'Quality' control menu is available for quality control and assurance functions. The 'Election Data Centre' enables the election officers to have a quick view of the current election, qualified candidates, registered voters and accredited voters.

The voter's dashboard (Figure 8) opens once a voter logs in with his/her unique credentials. The voter will only be successful in loging in if he/she is duely accredited. To be able to cast vote, the voter clicks on 'General', then on 'Voter Window' then casts his/her vote. The voter casts his/her vote by clicking on "vote" against a preferred candidate. To view the result, the voter clicks on 'General' menu option followed by 'Results' or click 'Election Statistics' on the voter's dashboard.

Of vital interest are Figure 9-11 which were used to satisfy the test cases specified in Table 1-3 respectively. The results of the test cases were as expected.

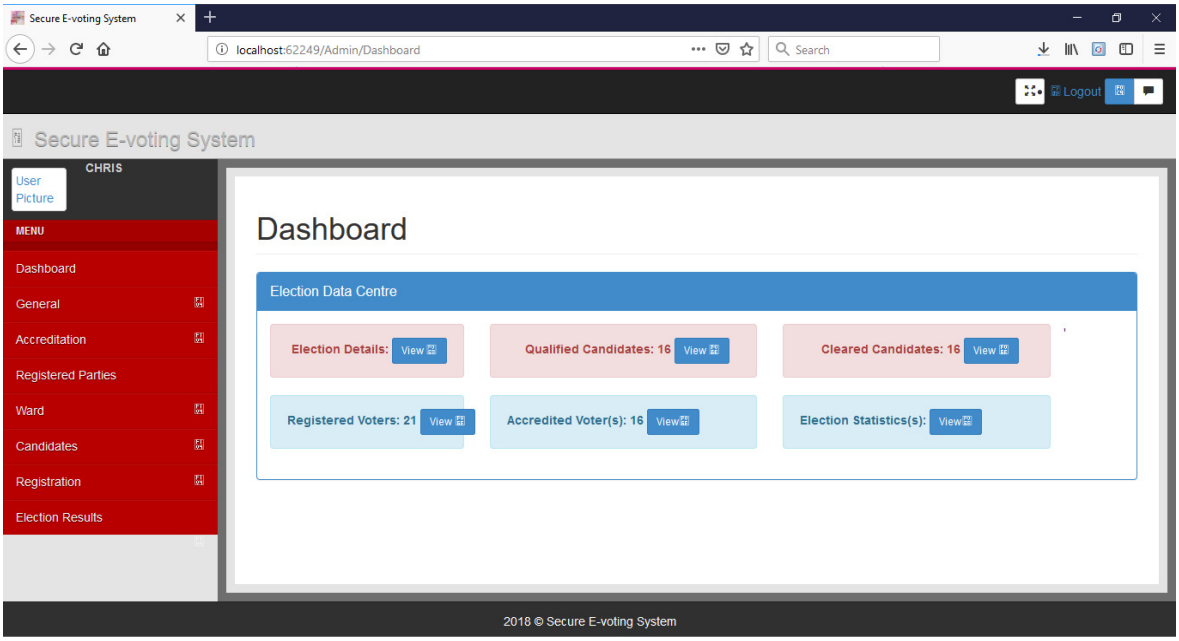

Figure 7: Administrative dashboard of the internet voting system

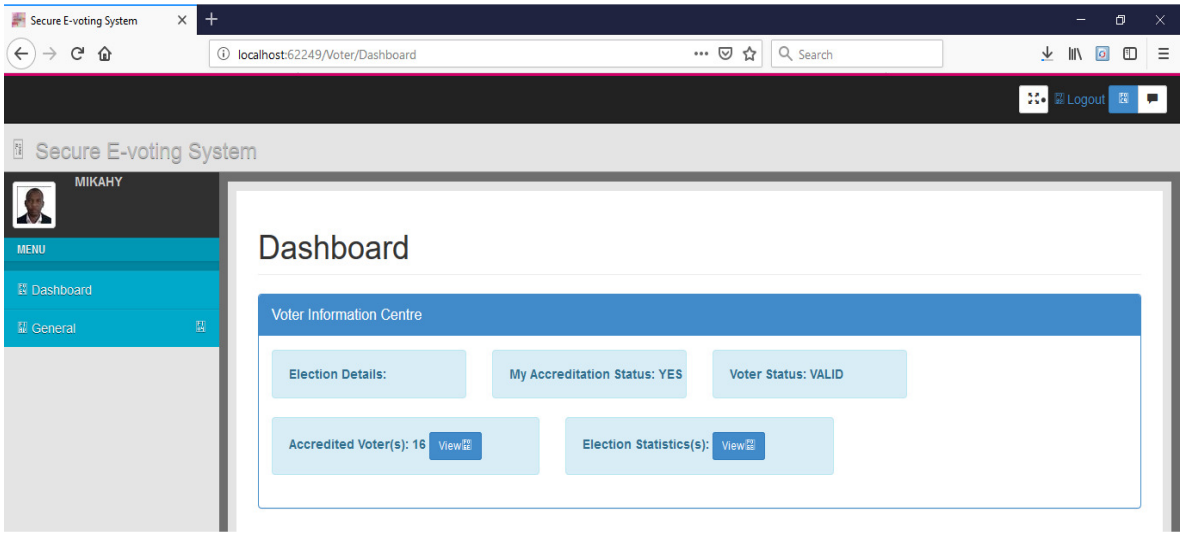

Figure 8: Voter dashboard

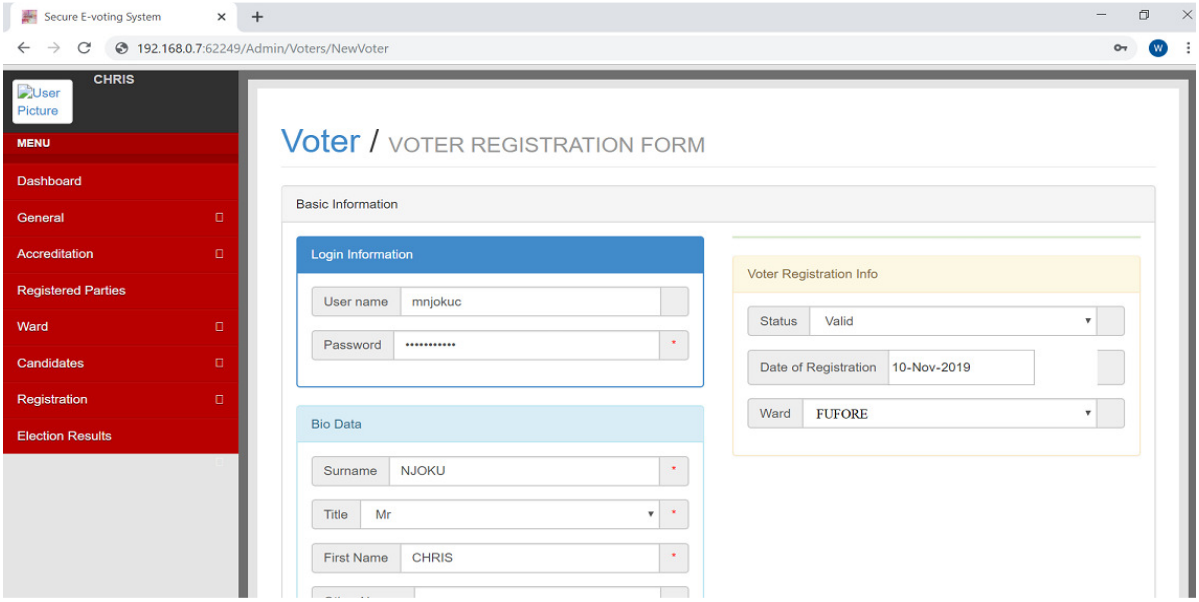

Figure 9: Voter registration test case window

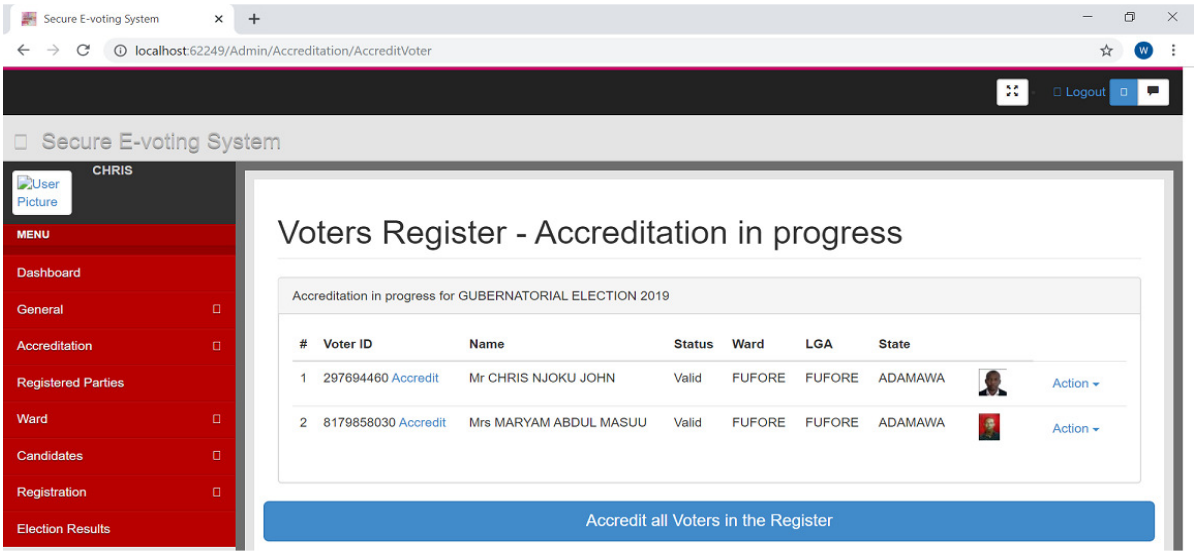

Figure 10: Voter accreditation test case window

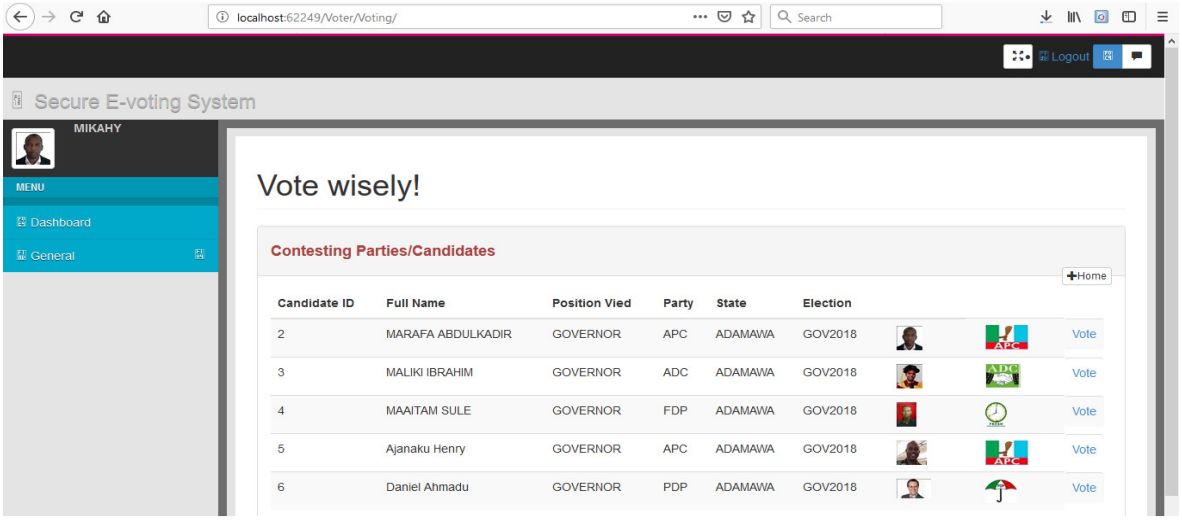

Figure 11: Vote casting test case window

3.2. Comparative Reconnaissance and Exploitation Testing

The two web applications are hosted on different machines on the same network. The details are as in Table 6. Note that the commands presented in Table 6 are not exhaustive. The Voter Login Exploitation Test Case was specified in Table 4. The results presented in Table 6 flowed from the execution of a penetration test aimed at revealing administrative logins on the both application streams. The LINQ-based i-voting platform (IVP) was hosted on a computer with IP address 192.168.0.21(host 1), and it's made accessible through http port 6224 whereas that of the LINQ-less IVP is hosted on a computer with IP-address 192.168.0.22 with application port number 1340.

Table 6: Vulnerability test case results

Parameters	Test machine	Host 1(LINQ-based)	Host 2(Mixed)	Result
IP Address:	192.168.0.7	192.168.0.21	192.168.0.22	Not applicable
Application link	http://{host IP}	192.168.0.21:6224	192.168.0.22:1340	Not applicable
Websploit command	use web/dir_scanner; set TARGET {host IP}; run; use exploit/autopwn; set TARGET {host IP}; run			Displays all directories on both hosts
SQLMAP command	sqlmap --url="http://{host IP}" --data="__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=%2FwEPDwULLTE4MjI5ODQ3ODhkZBhYr%2F8jkYBFxsKYA1YM1vPkqv5P%2FQj8KLA89PfymMCs&__EVENTVALIDATION=%2FwEdAARI43w1YsdHPRRITZvRBlVuY3plgk0YBAefRz3MyBlTcInkg%2Fut7Je4AtoEsfzZAOI85pbWlDO2hADfoPXD%2F5tdeqsY63Vwtk2NY2Vz7Ib0nYv%2BCWGPoIG6fglzvAXHKcM%3D&username=wilson&password=p%h7sjsdj&btnlogin=Login" -p username --banner			"not injectable" noticed on host1(see Figure 12) but displays the background database details including database server type; databases, database tables, and vulnerability to XSS attack. on host 2
	sqlmap -- url="{host IP}" --data="__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=%2FwEPDwULLTE4MjI5ODQ3ODhkZBhYr%2F8jkYBFxsKYA1YM1vPkqv5P%2FQj8KLA89PfymMCs&__EVENTVALIDATION=%2FwEdAARI43w1YsdHPRRITZvRBlVuY3plgk0YBAefRz3MyBlTcInkg%2Fut7Je4AtoEsfzZAOI85pbWlDO2hADfoPXD%2F5tdeqsY63Vwtk2NY2Vz7Ib0nYv%2BCWGPoIG6fglzvAXHKcM%3D&username=wilson&password= p%h7sjsdj &btnSubmit=Submit" -p username --sql-query="select name,master.sys.fn_sqlvarbasetostr(password_hash) from master.sys.sql_logins"			Result display the sa user account password hash on host 2 but not on host 1
SQLMAP command	sqlmap -u "http://{host IP}/Admin/Voters/BasicVoterInformation?voterid=89 --dbs"			Voter register exploitation with XSS vulnerability observed on host 2 and not on host 1

An SQLMAP command directed to both hosts produced an errors message that reveals the host database and account details on host 2(LINQ-less) whereas same operation was terminated on host 1 without such display. In the second test case involving accessibility to voter register, vulnerability assessment implicated the host 2 application on vulnerability to cross-site scripting (XSS) attack as against none on host 1.

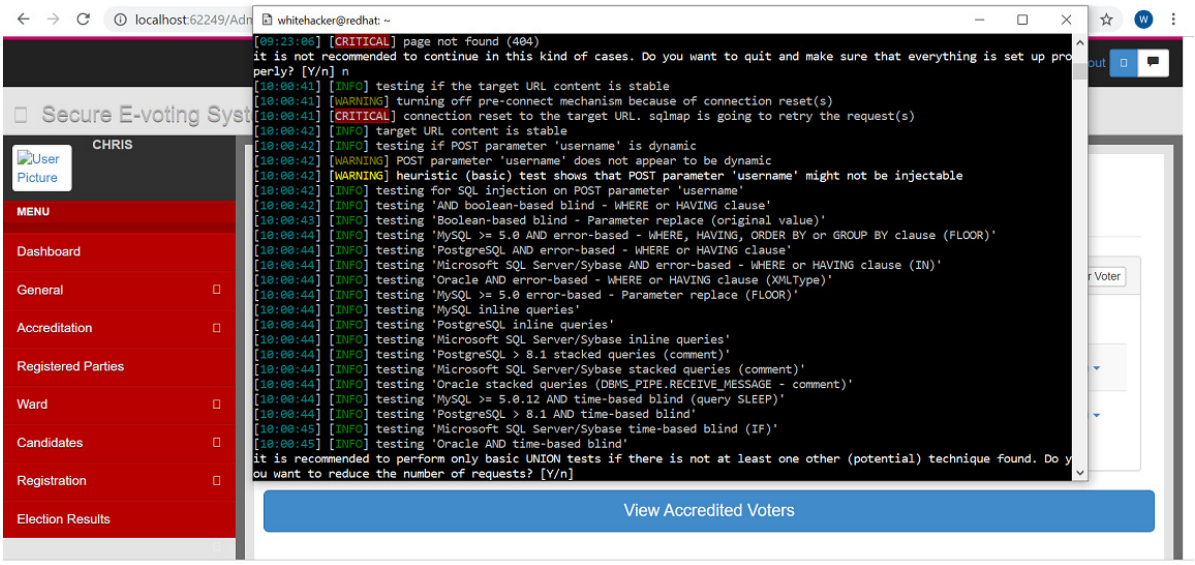

Figure 12: Voter login exploitation test failure on LINQ-based IVP

4. CONCLUSION

In this paper the comparative exploitation test results on two streams of same e-voting web application has showed that web applications built using carefully structured LINQs have the capacity to prevent SQL injection as well as well as XSS attacks. To ensure adequate application security, similar techniques should be adopted to curb other injections attacks.

5. ACKNOWLEDGMENT

The authors wish to acknowledge the assistance and contributions of the laboratory staff at Department of Computer Science, Abu Saleh College of Education Azare, Bauchi State.

6. CONFLICT OF INTEREST

There is no conflict of interest associated with this work.

REFERENCES

Al-Ameen, A. and Talab, S.A. (2013). E-Voting Systems Security Issues. *International Journal of Networked Computing and Advanced Information Management (IJNCM),* 3(1), p. 25.

Achieng, M. and Ruhode, E. (2013). The Adoption and Challenges of Electronic Voting Technologies within the South African Content. *International Journal of Managing Information Technology*, 5(4), pp. 1-12

Ahmad, S., Abdullah, S.A.J. and Arshad, R. (2015). Issues and Challenges of Transition to e-Voting Technology in Nigeria. *Public Policy and Administrative Research*, 5(4), pp. 95-102.

Alausa, D.W.S. and Akingbade, L.O. (2017). Electronic Voting: Challenges and Prospects in Nigeria's Democracy. *International Journal of Engineering and Science (IJES)*, 6(5), pp. 67-76.

Aranuwa, F.O. and Oriola, O. (2012). Improved Electoral Fraud Prevention Mechanism for Efficient and Credible Elections in Nigeria. *African Journal of Computing & ICT*, 5(6), pp. 7-10.

de Vuyst, B. and Fairchild, A. (2005). Experimenting with electronic Voting Registration: The Case of Belgium. *The Electronic Journal of e-Government*, 3(2), pp. 87-90

Ofori-Dwumfuo, G. O. and Paatey, E. (2011). The design of an electronic voting system. *Research Journal of Information Technology*, *3*(2), pp. 91-98.

Idike, A.N.A. (2014). Democracy and the Electoral Process in Nigeria: Problems and Prospects of the E-voting option. *Asian Journal of Humanities and Social Sciences*, 2(2), pp. 133-141

Javaid, M.A. (2014). Electronic Voting System Security. *Institute of Electrical and Electronics Engineers*.

Laukkonen, J. (2018). Which countries use electronic voting? Available at https://lifewire.com/which-countries-use-electronic-voting. Accessed on August 2019.

Mediayanose, O.E. (2018). The role of security in credible elections and sustenance of democracy in Nigeria. *Journal of Public Administration, Finance and Law*, 13, pp. 134-141.

Nwankwo, W. (2015). Development of A Model for Implementing risk Management Data Warehouse for Scanning Companies in Nigeria. Chelmsford: Anglia Ruskin University, pp. 13-14.

Nwankwo, W. and Ukaoha, K.C. (2019). Socio-Technical perspectives on Cybersecurity: Nigeria's Cybercrime Legislation in Review. *International Journal of Scientific and Technology Research,* 8(9), pp. 47-58.

Nwankwo, W. and Olayinka, A.S. (2019). Real-time Risk Management and X-ray Cargo Scanning

Document Management Prototype for Trade Facilitation, *International Journal of Scientific and Technology Research,* 8(9), pp. 93-105.

Nwogu, E. R. (2015). Mobile, Secure E - Voting Architecture for the Nigerian Electoral System. *IOSR Journal of Computer Engineering (IOSR-JCE),* 17(2), pp. 12-13.

Okoro, E. (2016) A cost-benefit analysis of Electronic voting operations and capabilities in sub-saharan Africa. *Journal of Business and Economic Policy*, 3(3), pp. 22-31.

Odunsi, W. (2019). Nigeria elections: International observer group gives damning report. Available at https://dailypost.ng/2019/02/25/nigeria-elections-international-observer-group-gives-damning-report/. Accessed on September 2019

Onimisi, T. (2015). The Prognoses of the 2011 Electoral Violence in Nigeria and the Lessons for the Future. *Mediterranean Journal of Social Sciences, 6(1)*. Rome: MCSER Publishing.

Reuters (2019). Observers report dozens killed in Nigeria election violence. Available at https://www.japantimes.co.jp/news/2019/02/25/world/politics-diplomacy-world/observers-report-dozens-killed-nigeria-election-violence/#.XYPj2ShKjIU. Accessed on 20 May 2019.

Rouse, M. (2011). E-voting (electronic voting). Available at http://whatis.techtarget.com/definition/e-voting-electronic-votingon. Accessed on June 2018

Uzedhe, G.O. and Okhaifoh, J.E. (2016) A Technological Framework for Transparent e-voting solution in the Nigerian electoral system. *Nigerian Journal of Technology*, 35(3), pp. 627-636.

Uwujiaeren, I., Okocha, C. and Orizu, U. (2019). Again, foreign observers knock 2019 General elections. Available at https://www.thisdaylive.com/index.php/2019/06/19/again-foreign-observers-knock-2019-general-election/ Accessed on June 2019.