Developing a fail-safe culture in a cyber environment using MySQL replication technique

Fatima M. Isiaka
Centre for Cyber Space, NSUK University, Keffi, Nigeria, and
Salihu Abdullahi Audu and Mustafa Ahmed Umar
Department of Computer Science, Centre for Cyber Space, Nasarawa State University, Keffi, Nigeria

Received 27 April 2018
Revised 11 June 2018
Accepted 16 August 2018

Abstract

Purpose – The dependence on the use of information systems for nearly every activity and functions in the internet is increasingly high. This form of interconnectedness has bolstered national economies, enhanced how governments interact with their citizens and how ordinary people connect with friends and family. However, this dependence has equally resulted to a high rise in vulnerability, threat and risk associated with more use of information and communication technology. Cyber-attacks that have the potential to disrupt or damage information system infrastructure are getting more complex with some level of sophistication. Traditional protection of information system infrastructure is no longer sufficient; systems have proven to be immune to failure or incidents. This paper aims to ensure that there is a continuous availability of services through a fail-safe proof.

Design/methodology/approach – MYSQL replication technique was used to develop a model based on three-tier layers using the principle of network interdependency and the replication techniques. Tier 1 depicts a Telecom organization serving as service provider that provides internet service to Tier 2 organization – a Bank; Tier 3 is the financial App that can be used by bank staff and customers. The fail-safe mode integrated mechanism enables Tier 3 to continue to render its services in the event of an attack on Tier 1 such as DDoS without disruption.

Findings – This technique succeeded in mitigating the loss of data if cyber incident occurred or reception of uninterrupted services is countered, which give rise to future master-to-master architecture.

Research limitations/implications – The study conducted is limited to the design and development of a fail-safe system for interdependent networks or systems using MYSQL replication technique.

Originality/value – In an interdependent environment such as the cyberspace, the sectors are interdependent for optimal results. The originality of the work ensures that there is availability of services which is sustained and that data integrity is assured using the fail-safe technique based on MySQL replication method.

Keyword Mitigation

Paper type Research paper

1. Introduction

The cyber environment, which is more popularly known as the cyberspace has various accepted academic definitions presented by various academic scholars, individuals and

International Journal of Crowd Science
Vol. 4 No. 2, 2020
pp. 149-170
Emerald Publishing Limited
2398-7294
DOI 10.1108/IJCS-04-2018-0008

organizations. According to the U.S. Department of Defense, the cyberspace can be defined as a global domain within the information environment consisting of the interdependent network of information technology infrastructures, including the internet, telecommunications network, computer systems and embedded processors and controllers? Although different individuals and organizations have different or more preferred definitions of the term, it is safe to say that their various definitions share the fact that the cyberspace is made up of computers, networks, users, devices, electronic transactions, transit data, processes as well as hardware and software (CERT UK and GCHQ, 2012; Chakraborty, 2009; Bishop, 2003; Gollmann, 2005a, 2005b, 2005c; Gollmann, 2005a, 2005b, 2005c). In the last couple of decades before the development, emergence and adoption of the internet, the cyberspace was inexistent. Thus, elements such as devices, networks, data storage and transportation as well as telecommunications were not possible as all day-to-day activities were carried out in the usual traditional approach whereby events had to be performed or carried out physically such as sending of messages from a party or destination to another, transacting in business and services, saving or sharing pieces and bits of information and communicating between two or more parties.

In the past couple of decades before the development, emergence and adoption of the internet, the cyberspace was inexistent. Thus, elements such as devices, networks, data storage and transportation as well as telecommunications were not possible as all day-to-day activities were carried out in the usual traditional approach whereby events had to be performed or carried out physically such as sending of messages from a party or destination to another, transacting in business and services, saving or sharing pieces and bits of information and communicating between two or more parties. During the early 1990s, the internet was invented.

Three decades on, the field of computing and information technology has witnessed arguably one of the fastest growth in human history with devices and concepts reaching out to vast majority of the world while been continued to be adopted on a massive scale from simple individual and corporate activities to the running of governments (Mbanaso *et al.*, 2015; Arshad and Matt, 2009; Locked, 2014). Unlike before the information age, virtually all day-to-day activities of individuals and organizations nowadays are carried out on a very huge scale using computers, users, networks, telecommunications and data that together make up the cyberspace. Therefore, it is important to note that the cyberspace does not just serve as an environment that helps ease our daily activities around the world but also as a necessity for making our activities easier, faster and more accurate.

The transformation to digital age has seen a lot human activities change in terms of their execution and operational procedures due to a lot of reliance on technology as well as the connection between many components. Nowadays, the survival and strength of every national economy across the globe depends on the status of these core sectors i.e. the power or energy, banking and or finance and the telecommunications sector which exclusively depends on one another. These three critical interdependent sectors must be kept running all day, week, month and year round because, the collapse of any of these sectors usually has a cascading effect to the two other sector as each sector absolutely depends on each other for their individual operation (Leidigh, 2005; Leidigh, 2005; Greene, 2012; Gollmann, 2005a, 2005b, 2005c). For instance, the loss of power in the power and energy sector would render the banking and finance and the telecommunications sector useless, and as such, these would lead to the disruption of flow of money, valuables, assets and information, without which no activities can be planned, managed and executed in any case. In another instance, the loss of the banking and finance sector would lead to failure of the other two (although, more slowly). Furthermore, the loss of telecommunications would mean loss of flow of valuable and live or up-to-date information which is the most critical pillar of operation for the other two sectors and every other economic or national sectors and sub-sectors. Thus, it is important to ensure that all three sectors are kept up and running at every point of every year as all other sectors and activities of any nation depends on them, especially the telecommunications sector, which is responsible to connecting all national activities together. In a situation whereby systems or organizations depend on each other for survival and operations, it is referred to as network interdependency in computing. Therefore, with the advancement and high quality research going on in various fields of computing, various types of systems and concepts are designed and applied to protect critical sectors and infrastructure from complete shutdown. These designs are prepared to ensure that should any of these systems fail, they would do so in a more secure manner while getting them back up and running within the shortest possible time period.

1.1 Statement of the problem

The cyberspace is considered one of the most important aspects of daily human activities as it offers enormous services that make our activities faster, cheaper and more accessible. It faced with various existing and potential security challenges as more sensitive information about the users, devices, networks and other components are been uploaded and transferred from one point to another within the cyber environment. Hence, the security of the cyberspace and its components must be strongly considered to protect the confidentiality and integrity of the users as well as the overall availability of the system. Therefore, this work presents a developed fail-safe culture for cyber environment; a system that is designed to help in mitigating the impact of resource loss as well as the confidentiality and integrity of the users and devices in the case of a system breach or failure.

1.2 Research questions

The following are the questions addressed by this work:

Q1. What happens in the banking sector when the primary server is attacked?

Q2. Is it cost-effective for each organization to store their data locally?

Q3. Is it sufficient to use all available resources to secure one data without planning for its eventual failure?

Q4. What happens to banks customers when system availability is breached?

1.3 Objectives

The aim of this paper is to develop a model for ensuring the secure failure of a cyber-system within the cyber-environment, with objectives to mitigate the impact of incidents within the cyber-environment, maintain the confidentiality and integrity of cyber resources in the event of cyber and minimize the cascading effect of system failure spreading to other critical sector. In an interdependent environment such as the cyberspace where the sectors are interdependent optima results, the work will ensure that availability of services is sustained and data integrity assured. This study is limited to the design and development of a fail-safe system for interdependent networks or systems using MYSQL replication technique.

2. Literature review

The importance of fail-safeness of systems cannot be over-emphasized as the omnipresence of cyberspace systems development and deployment into consumer-oriented products and services (e.g. PDAs, mobile phones, safety-critical systems such as avionics and car engine control) continues to increase. With our increasing reliance on the use of technological systems on a day-to-day basis around the world for series of complex activities ranging from transportation, banking, generation of energy and power to the running of government, comes along a series of possible and potential challenges regarding the use of these systems, among these issues are faults and security of critical system. In a situation whereby the security of any of these systems is breached, so many users, resources and information are endangered. Therefore, there is every need for fail-safe system in virtually every cyberspace technological system, most especially in safety-critical systems such as the transport systems, information systems, network systems and services. According to (Jack, 2011; Gollmann, 2006; Gollmann, 2005a, 2005b, 2005c; Ida *et al.*, 2013; Locked, 2014; Sajal *et al.*, 2015) fail-safe system is the ability to create a system that is safe-to-fail, on the other view, a fail-safe system can be defined as a system that is designed to consist of a shadow system (a secondary system) whose primary function is to monitor and take actions on the primary system in the event that the primary system security is been compromised by attackers or its physical components are faulty. The primary system carries out all the major activities it was specifically designed for while the secondary system carries on with the monitoring and sensing of irregularities within the main system to avoid getting caught off-guard by attackers and possible system component failure. Typically, in the event that the main system is breached and the secondary system does not respond, high-level damages would be dealt on not only the system, but also the information contained in the system such as data of users, network and connectivity, systems, services, resources, telecommunication channels and so on. Thus, the implications of complete failure of the overall system means loss of confidentiality, integrity and availability of the system, which could lead to loss of properties and other negative impacts on the concerned users.

Although there exists various designs and types of fail-safe systems depending on the technological system in question, each of these systems is designed to serve the same purpose i.e. respond to a fault or breach and mitigate the amount of resource loss as well as protect the confidentiality and integrity of the system and its contents within the fastest possible time. There exist a number of fail-safe systems designed in the last couple of decades. A typical example of these systems includes a fault tolerance system for controlling the speed of a train called SACEM (Guiho and Hennehert, 2000; SANS, 2002; SANS INSTITUTE, 2004; Semantec, 2011), their systems which emerged as an automatic speed control system in the field of railway systems, the strength of this system is that, it function best in a rail system, constantly monitoring the critical part such as the breaking system because failure in the breaking system always resulting to a catastrophe, their system do not take into consideration the interdependency of critical unit in a cyber-environment whose effect always cascade to other dependent unit.

2.1 Fail-safe system architecture

The architecture of almost any system physical in the form of hardware or nonphysical in the form of software is designed based on the functional requirement of the overall system. In the area of fail-safe system design, there exist different types of architectural models and solutions used to ensure safety. These system designs range from the simplex and duplex architectural designs to the dual duplex and TMR architectural models. Each of these architectures considers the availability of the system firstly before considering the issues of its safety. A typical example of systems that make use of these architectural designs includes the electronic interlocking railway system most commonly known as the ELEKTRA. This system is designed using two TMR channels with one of the channels mainly focused on controlling/monitoring the system's default configuration and the other ensuring that the safety conditions of the system are respected and kept intact. Thus, the availability and reliability of the system are achieved through the use of live triplicated hardware components in each of the channels while the inter-channel checking helps to achieve system safety.

In an effort to minimize the amount of downtime experienced, Oracle designed a fail-safe system for its databases as well as other software applications that run on the Microsoft Clusters and are configured with the Microsoft Cluster Server as shown in Figure 1. The system works with the Microsoft cluster server to configure both the software and hardware resources for high-level availability, as presented in Figure 1, the system is designed alongside another system (a shadow system) each with a set of cluster disks and RAID array. Once the system is configured, the numerous nodes contained in the cluster appear to the clients and end users of the system in the form of a single virtual server while allowing them connect to a single and fixed network address known as the virtual address without the need for any knowledge about the underlying cluster. If in any case, one of the nodes in the cluster loses its availability, the Microsoft cluster server transfers the workload of the unavailable cluster, which may also contain the client requests, to another node, that is available. For example, on the left hand side of Figure 1, it shows a two-node cluster setup where both nodes are fully available and active. On the system surface, this setup looks no different from setting up dual-independent servers; except that the subsystem responsible for storage is setup such that the disks are physically connected to the two nodes via a shared storage interconnect. Although the two nodes are connected physically to the same disks, the Microsoft cluster server ensures that a disk can be owned and accessed by only one node at any given time (Steve, 2015).

On the other hand, the right-hand side of Figure 1 displays how, whenever a certain software or hardware becomes unavailable on one node, its entire workload is automatically

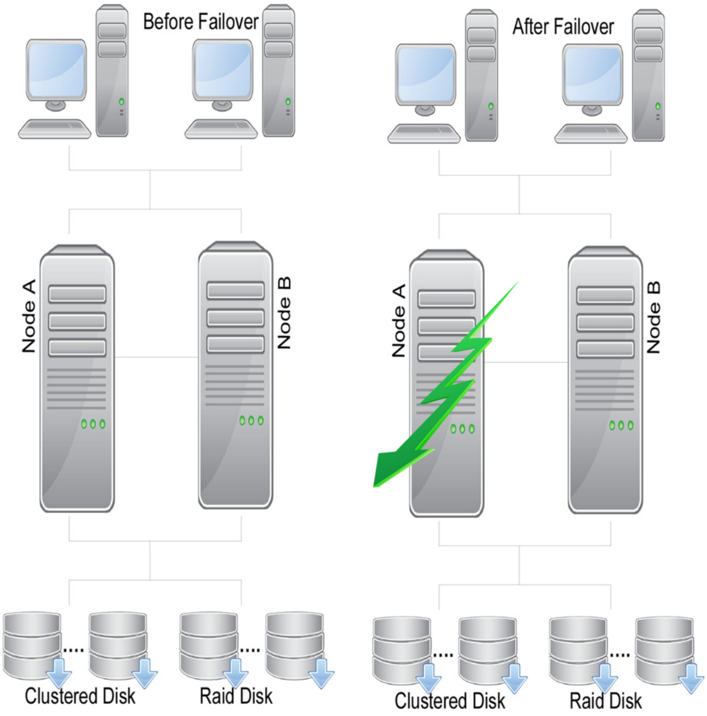


Figure 1.
Oracle fail-safe architecture
(Steve, 2015)

moved to the active node and restarted, without the intervention of the system administrator. During the failure, the ownership of the cluster disks is completely released from the unavailable node (A) and then picked up by the active node (B). If even an instance oracle database was running on the first node (A), then the oracle fail safe system will restart the database instance on the second node (B). Thus, clients can then have access to the database through the second node (B) using the same virtual address that was used to access the database when it was initially hosted by the first node (Figure 2).

3. Security risk and analysis

Computer and network security involves the monitoring, analysis, detection and prevention of unauthorized access to cyber systems by illegitimate users resolve. It also involves the protection of the cyberspace against activities such as the intentional disruption or diversion of systems and their resources as well as the integrity and confidentiality of the information stored or passed through the system. Additionally, the concept of network and computer security is not limited to only the software components of the system as it also consists of controlling user access to the physical system such as the hardware. However, to ensure that all these activities of monitoring, controlling, detecting and preventing against unauthorized access to a system are fully followed up and implemented, the principles of system security architecture must be duly followed. Over the years, the need for computer and network security has become increasingly important. According to a community emergency response team (CERT) (CERT UK and GCHQ, 2012; Semantec, 2011; Willian, 2015; Zahri and Syahrul, 2010) survey taken by the Carnegie Mellon University (a Centre for Internet Security Expertise) from the year 1998 to 2003, there has been over 400 increase in the number of reported system security incidents. Hence, this means that more robust mechanisms and approaches are needed to mitigate the impacts of potential loss for organizations and individuals; one of these mechanisms and approaches is the concept of fail-safe. Therefore, after all design decisions of any cyber system have been clearly outlined, the concept of a fail-safe system can be coupled together with tighter organizational and individual security policies, mechanisms and principles.

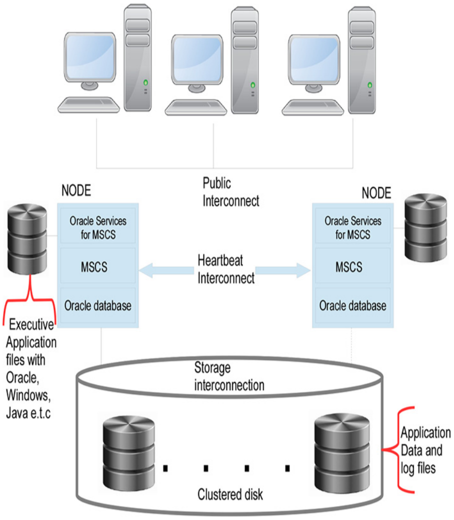


Figure 2.
Architecture of a configured Oracle fail-safe hardware and software components (Steve, 2015)

4. Cyber security assessment

To prepare a sufficient and efficient defence mechanism on a cyber-system so as to prevent or mitigate against potential cyber-attacks, it is highly necessary to measure the threshold of the attack as well as how prepared your defence is and what it is prepared for, should the attack be successfully launched against the system. The cyberspace security assessment provides vital information about these possible events by using assessment factors such as the vulnerability, threat and risk analysis (FFIEC, 2015).

4.1 Vulnerability assessment

The primary purpose of carrying out a vulnerability assessment is to use what was observed and identified during the information gathering stage and perform a test to determine current level of system exposure. In addition, it provides other vital information, e.g. whether the current defence mechanisms are sufficient enough to withstand or mitigate threats and attacks against the confidentiality, integrity and availability of the overall system. There currently exist various tools and techniques that can be used for the identification of certain system vulnerabilities such as the Nessus, SAINT, Whisker and Sara tools (Ken, 2003). The vulnerability assessment stage also deals with the area of system penetration testing with the main objective set at extracting valuable information in the form of either a text, password or even classified document. After a full-phased penetration testing has been carried out, the specific vulnerabilities observed, are graded based on the individual levels of risk they internally or externally pose to the system. Thus, a low rating could be applied to vulnerabilities that are low in severity and exposure while those with high rating in severity and exposure are considered of high vulnerabilities.

4.2 Threat assessment

According to Ken (2003), a threat can be defined as any object or entity that is capable of contributing to the destruction, tampering or interruption of any system or service that is of value. The primary objective of conducting a threat assessment is to measure all or any possible elements that could be a risk to a cyber-system. Threats are majorly categorized into human and non-human elements such as hackers, theft, backup operators, technicians and flood viruses, fire, electrical faults, heat controls respectively. In relation to the system's environment of deployment, identified threats must be properly assessed to determine their effects on the system. Additionally, threat and vulnerabilities go hand-in-hand and as such, can be measured and graded in a similar method such as in terms of capability and impact as well as motivation.

4.3 Risk assessment

One of the final and most important phases in the cyberspace security assessment is the risk analysis. The risk analysis of any system is performed to assess whether or not the currently employed organization policies, protection and procedures put in place are sufficient and adequate enough to defend the system against cyber-attacks. In a situation whereby the safeguards put in place do not provide adequate protection, the system can be considered vulnerable. Hence, one way to determine that is to perform an analysis and review of the currently existing and planned system defence mechanisms to determine whether or not the previously known and discovered threats and risks have been mitigated. Additionally, with the activities of assessment and review of the current and existing system defences, risk analysis provides information about the consequences faced by the system in the event of a system breach as well as other information about how to generally mitigate existing or potential risks.

4.4 Contemporary cyber-attack

Due to the boundless opportunities and devices involved in modern day technology as well as the fact that these technologies have no defined or set form, the cyberspace is currently faced with various types, mediums and forms of attack. Also, with the surging growth in the population of cyberspace users and devices comes new cyber-attacks on a daily basis around the world. Unlike the earliest known cyber-attacks, contemporary cyber-attacks are much more complex to monitor, detect and mitigate due to their compounded nature. Figure 3 showed a graphical representation of the attacks that have contributed to the largest amount of loss since 2005 as investigated and presented by the computer crime and security survey. According to the survey, the data presented suggests that virus attacks are the most commonly carried out attack across the cyberspace. Other common attacks include the malware attacks, spyware, worms, password attacks (e.g. Brute-Force attacks and Dictionary Attack), Denial-of-service attack (DDoS) (e.g. execution through buffer overflow, teardrop, Smurf attack or physical disconnections), Man-In-The-Middle (MITM) Attack and Trojan Attack.

All of these attacks are mostly carried out with certain motives such as political, business, social and economic motives. Additionally, each of these attacks aims at disrupting the triad and goals of network security as they are targeted against the confidentiality and integrity of user data or a particular system and its entire availability. Regardless of whether or not an attack is targeted, or an attacker is making use of tools specifically built to carry out an attack, most cyber-attacks share a certain number of similar patterns and stages (Larman, 2005; Zahri and Syahrul, 2010). While at this, some of these cyber-attacks end up meeting their primary objective while some eventually get blocked along the line of system defense. An attack may be designed and carried out in form of repeated stages, most particularly if it is done by a persistent adversary. The cyber-attacker continues to probe the defense of the system to obtain a weak point on the system. Thus, if the system is not properly guarded and becomes exploitable, the goals become less close to getting achieved. These are other problems faced is what the paper is aimed at addressing.

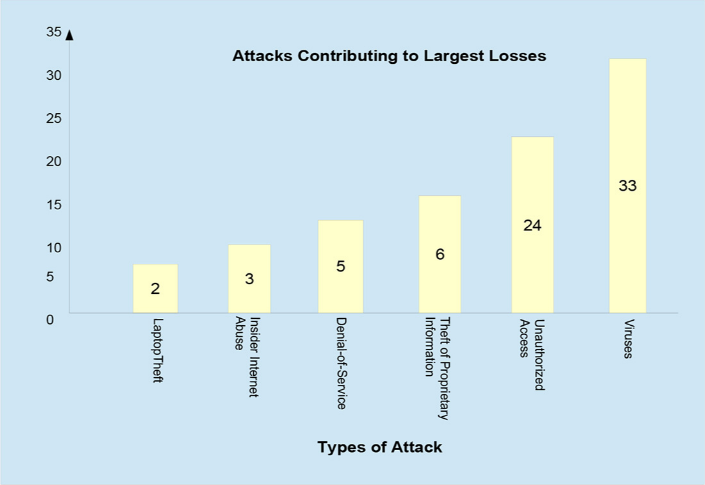


Figure 3.
Bar chart of attacks with the largest lost contribution (Gollmann, 2005a)

5. Methodology

The purpose of the method used is to provide a structure for the overall study by showing how various parts of the study work, to address the primary problems. Figure 4 depicts the design framework for the Universal System Model. However, this paper considered several philosophical thoughts; it favored the mix paradigm (qualitative and quantitative approach) because of the characteristics of the empirical variables and focused mainly on the following components:

(1) *Environment*: the environment indicate where the system can be used, these include the following:

(2) *People*: These include bank's staff and customers that can use the system, including the system administration.

(3) *Organization*: Although the case study of the work is banking sector and the Telecommunication, other organization such as e-commerce and education.

(4) Governance can be integrated into the system.

(5) *Technology*: This emphasis on the technology that will be deploying in using the system as well as in designing the system. These include(s) Netbeans IDE, JavaFX Scene Builder, MYSQL Workbench, etc.

(6) *Strategy*: This look at the case study used for the work, a model of a Fail-Safe is design that can be used in a cyber-environment, such environment is characterize by interdependency of critical component of economy for optimal result.

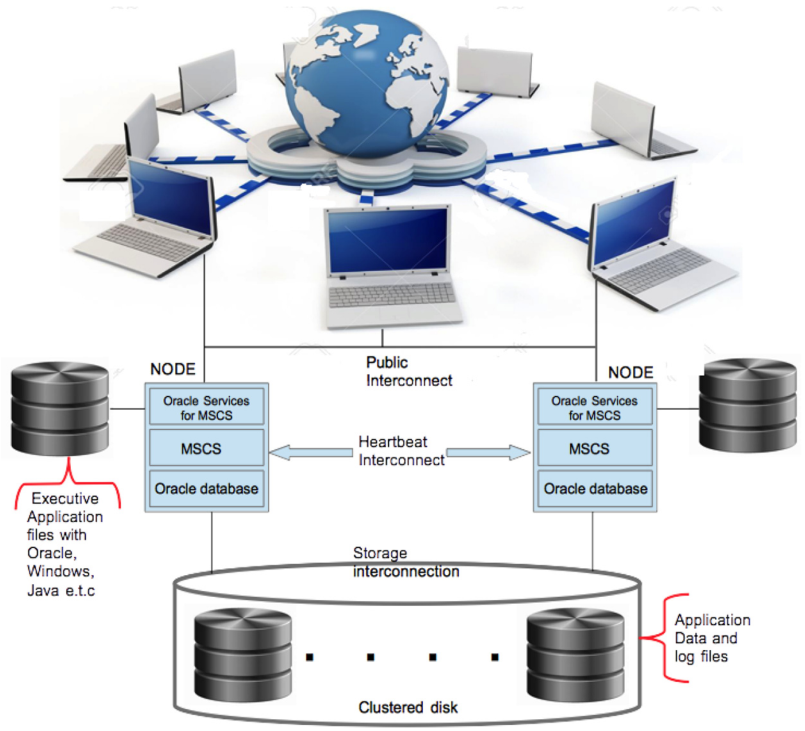


Figure 4.
The design framework for the universal system model

(7) *Methods/technique*: The methods/technique use are highlighted as follow: Documentation: The study used a well-documented banking transaction procedure

(8) *Develop/build*: The model was develop using the available technology as listed in the framework

(9) *Validation*: The system is validated using the following procedure:

- *Test 1*: Both the primary and the secondary server are working correctly i.e. replication processes is working correctly.
- *Test 2*: The primary server is attacked (DDoS), the secondary server takes over control in real time.
- *Test 3*: The primary server is restored, the secondary server update the main server of the processes that takes place while it was down and the process continue.

(10) *Methodology*: The mix paradigm was carefully chosen because of the nature of the empirical variables used for the work.

(11) *Philosophy*: The pragmatic approach was used because of the fact that both external and multiple view was chosen to best answer the research question.

The main setting adopted for the entirety of this project is based on a general University environment such as the offices, laboratory services and the library. The resources used to aid the development and successful completion of the project include a working personal computer system, programing tools such as software applications and Web platforms, journal articles and other scientific publications, as well as textbooks and other online resources.

```
Require: filteredData = Predicateactivity
Ensure: newValue = null
  Return null
  String: lowerCase f ilter ⇐ newValueLowerCase
  CustomerID ⇐ newdata
  while filtertext ⇐ empty(0) do
    Display − all − persons
    Display − all − personsID
    if activity is lowercase
     lter then
       return ⇐ true × filter − matches − firstname
       customerother ⇐ (customer)object          ▷customer
         is set to object
    else if String − activity = get − account number then
       Customer ⇐ model × filterdata
       CustomerTable ⇐ sortedData
       CustomerList ⇐ sortedData2
       Return ⇐ false
       Return ⇐ true
       refreshdata
       refreshtable
    end if
end while
```

5.1 Model development

In this work, the fail-safe system development consists of three main phases, including the component, component analysis and the Data Centre. The component of the system was developed using the NetBeans IDE and JavaFx Scene Builder software application platforms, the test client is developed, which is referred to as the BankApp discussed in the next chapter of this work. At the Data Center, the Fedora Linux operating System is run on a VMware workstation to allow virtualization and real-time synchronization of information (including updates and modifications) from the primary server to the backup server without affecting the production load and availability of the main system or increasing the impact of the backup on the main system. This is achieved using a technique known as MySQL replication. Additionally, the MySQL Server is used as the database management system within the data centre while the Snort IDS (Intrusion Detection System) is used for monitoring and detecting network probes and attacks. Furthermore, on the analysis component, the MySQL Workbench is used for the design and management of the entity relationship diagram (ERD), which presents the relationship between the various existing entities and their instances within the model to make the implementation of the information structure possible within the database.

```
Require: inthash = 0
Ensure: hash = null
  Return hash
  newValue ⇐ ture
  String: lowerCase filter ⇐ newValueLowerCase
  N ⇐ n
  while filtertext ⇐ empty(0) do
      Displayallpersons
      Displayallpersons
      if object is Staff then
          return ⇐ false×
          staff other ⇐ (staff) object                ▷staff is set to
            object
      else if Stringactivity = getaccount number then
          Customer ⇐ model × filterdata
          if personID is null then
             other.personID ⇐ null×
             staff other ⇐ (staff) object
          else if personID = other.personID number then
             Return ⇐ false
             Return ⇐ true
             bankapp2 ⇐ personID + otherstaff
             refreshdata
             refreshtable
             refreshtable
          end if
  end if
end while
```

5.2 Tools and techniques for design

Typically, in the areas of information and computer system design and development, one or more programing language(s) is required for the development of the system (e.g. the front-end, Interface and back-end component design). Some of the languages used for the front-end component of the system include Java and JavaScript, C++, PHP, HTML, Python, IOS and so on while those used for the backend (Database) component include the SQL, SPARQL, WebDNA, dBase and WebQL. Each of these languages is mostly used within a software development platform such as the notepad, visual studio, Dreamweaver, NetBeans IDE (Integrated Development Environment) and the MySQL workbench.

At the fail-safe system development process, the primary programing language used is Java, precisely the Java 8 from the Java Platform Development Kit (JDK). This is because, it is highly object-oriented and as thus, allows the creation of modular programs and reusability of codes while supporting platform independence, flexibility, ease of writing, compiling and debugging. Additionally, the jfoenix library is used within the JavaFx Scene builder (part of the Google Material Design Library) to build and customize the test client's graphical user interface. Furthermore, the Java API (Application Programing Interface) is used to assist in the use of java packages, classes and interfaces required to successfully carry out development within the JDK while the DAO (Data Access Object) and STP (Single Turn Pattern) are used to provide database access to information stored in a different location and manage user session by restricting and ensuring that only one instance of a certain class exists on the java virtual machine respectively. Finally, the JPQL (Java Persistent Query Language) as part of the Java Persistent API (JPA), is used for making queries against the entities that are stored within the database at the data center.

Using the concept of network interdependency in cyber environments, it can be seen that three key national and economic sectors depend on the telecommunications sector to fully function at any particular time with each of them having their sub-sections of activities going on such as the financial activities that are performed in the banking and finance sector e.g. transactions between the bank and its customers. To illustrate how the fail-safe system is associated with the activities in Tiers 1, 2 and 3 as shown in Figure 5, a test client (BankApp) is developed using Java programing language in the Netbeans IDE.

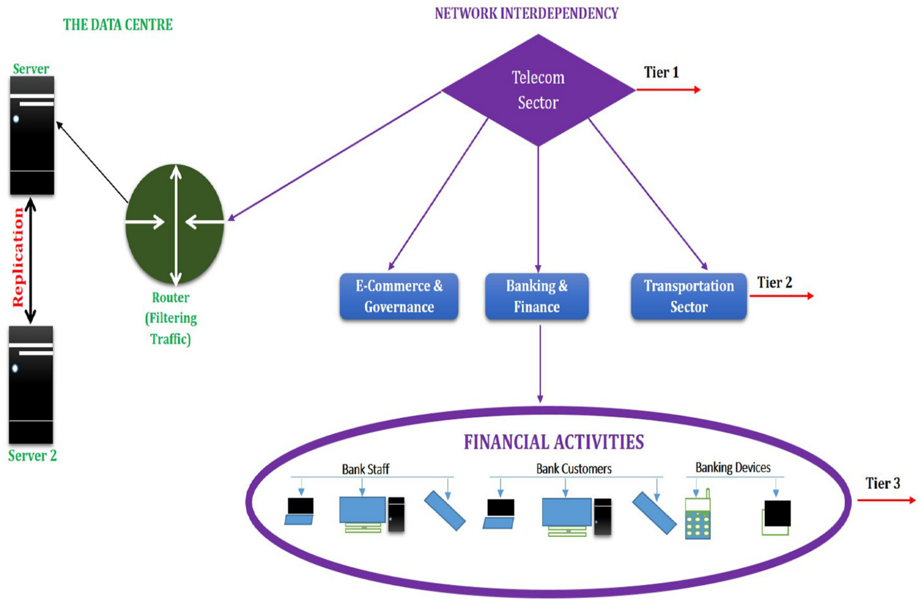


Figure 5.
Architecture of the fail-safe system

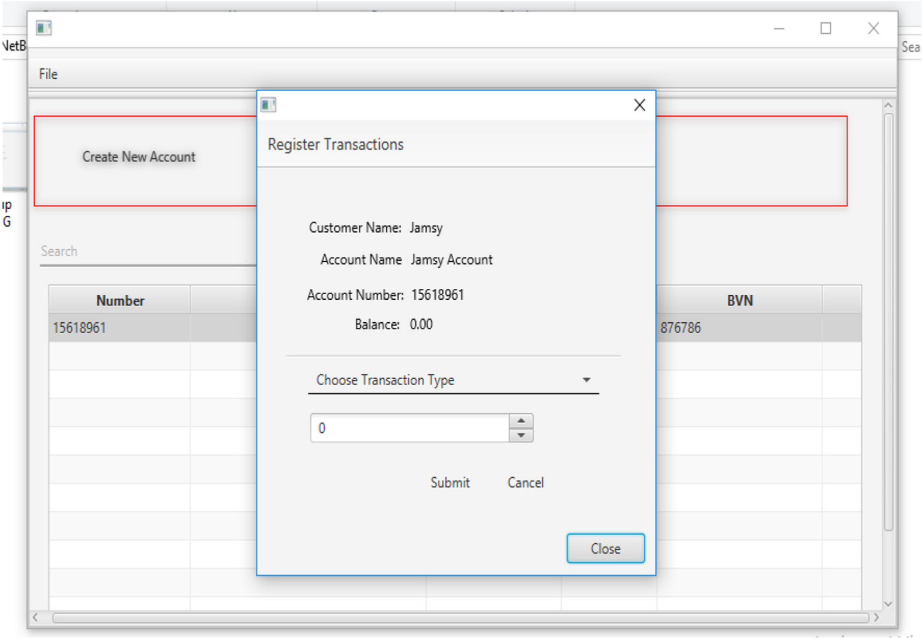
5.3 The test client (BankApp)

The test client was developed as an interface that is used on a daily basis by the bank to carry out their transactions and other related activities. It consists of three main components namely the front-end, backend and GUI component. First of all, before the front end or GUI was developed, the backend was developed to serve as a placeholder of information passed through from the front-end and GUI components. Using an ERD as shown in Figure 7 to illustrate the logical structure of the bank's databases, three basic elements of the ERD were maintained (i.e. the entities for which the bank wants to store information, the attributes the bank wants to collect an entity's information for and the relationship between the entities.

Second, at the front-end component of the BankApp, various banking activities are performed such as the registration, removal and suspension of staff by the bank's administration as well as the registration and conduction of banking transactions by the bank's staff (Figure 6). Third, at the GUI component, some of the nonfunctional requirements of the BankApp are developed and maintained using tools and packages such as the JavaFx Scene builder, which consists of the Google material design library. In summary, after the development of the BankApp, all banking activities on the system require the transmission and storage of information on the databases of the bank. These databases also need to be stored and protected on a server. Hence, a bank server is created to host these sets of data and databases on a virtual machine as shown in the next section.

6. The fail-safe system design and development

Universal System's Model is used to design the fail-safe system based on the core elements as shown in Figure 5. First, the goal of the fail-safe system is to ensure that, in the event of any security breach or action that leads to the unavailability of resources such as access to data-center by legitimate users, a secondary system is able to carry on the functions of the



Register Transactions

Customer Name: Jamsy

Account Name Jamsy Account

Account Number: 15618961

Balance: 0.00

Number	BVN
15618961	876786

Figure 6. The BankApp interface

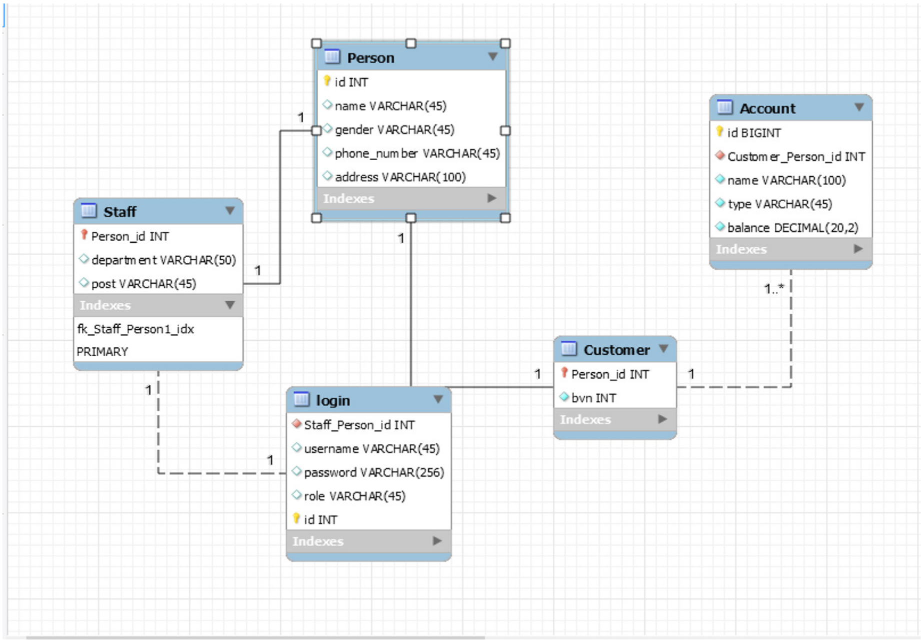


Figure 7.
The entity relation diagram of the BankApp

primary system without the end-user even noticing any changes or interruptions during system usage, hence drastically mitigating the risk of resource loss. Second, the input taken by the fail-safe system in this study is the data generated in the test client (BankApp). This includes any form of information inserted, modified or deleted from the test client such as customer bank account and transactions details (as in the case of a BankApp). Third, at the process element of the USM, the primary process that occurs within the fail-safe system is the technique of replication, which occurs at the virtualization section of the datacenter where information sent or modified on the primary server is replicated in real-time unto the secondary data. Fourth, the main output expected from the fail-safe system is the replicated data taken at the process element by the backup server from the primary server in real-time. Finally, at the feedback element, the fail-safe system is evaluated to determine whether or not the fail-safe system is reliable. For example, if the primary server or system becomes unavailable due to reasons such as denial-of-service attack, the fail-safe system is tested to determine if the backup server can immediately take over without any interruptions at the test client side (BankApp).

The entity relation diagram (Figure 7) shows the various object in the BankApp and it is associated entities and the number of character and the type associated with each character and the use-case diagram (Figure 8) shows the objects in the system, for example, how the admin and a staff can use the system; the admin in this case can register a person or suspend a staff if necessary; the staff can register customer and conduct bank transactions as well.

7. The data center

By considering the size and sensitivity of the data processed on a daily basis by banking institutions, there exists the need for a warehouse for these large chunks of data. This is known as the datacenter. It serves as a facility for organizations such as banks where

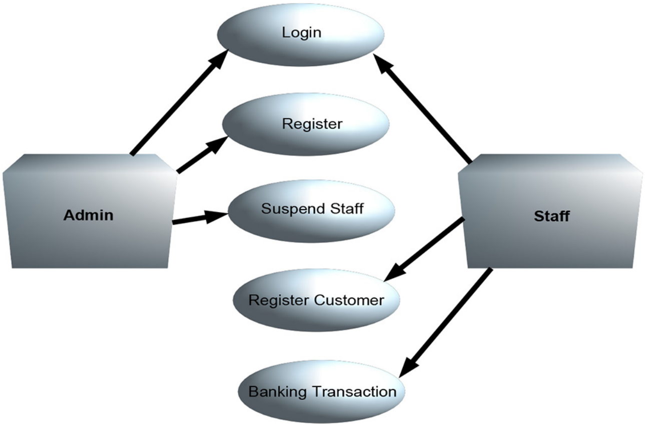


Figure 8. The BankAPP use-case diagram

information is stored, managed and disseminated in a centralized manner. Consisting of two major servers, namely, the primary (main) and secondary (backup) servers also known as the Master and Slave Servers respectively, the datacenter for the bank was virtualized. At the datacenter, real-time data received at the Master server is consistently replicated onto the Slave server in real-time. Additionally, there exist sub-servers on each of the two main servers, responsible for keeping information up-to-date and backed up between the two servers as shown in Figure 9.

At any point, should the Master server get shut down or become unavailable due to circumstances such as security breach, the Slave server automatically takes over operations from the failed master server without the end-users noticing any change or interruptions while the sub-servers of the slave server begin to replicate information received by the new master server until the failed server is restored to a normal condition. Hence, in any information system, the more the number of master servers, the higher the chances of sustaining the availability of the system, which would in turn maintain the integrity and

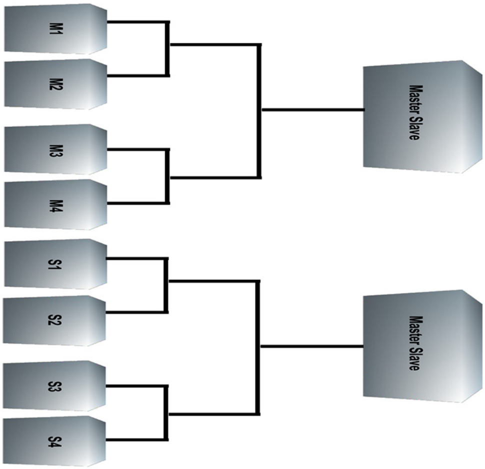
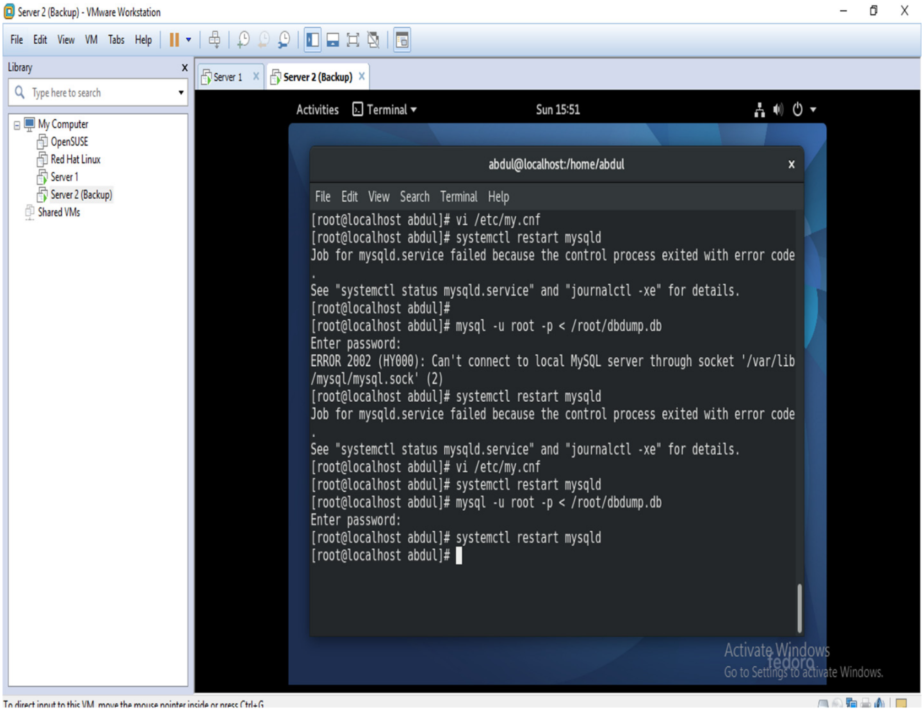


Figure 9. The BankAPP use-case diagram

confidentiality of the information contained within the servers. Furthermore, before information gets into the Master server from the BankApp in the first place, it is passed through a snort Intrusion Detection System, which performs scanning and filtering of packets to reduce the chances of attacks against the main server. The Snort IDS is placed between the main source of the information (e.g. the BankApp platform) and the virtualized servers.

For a system, especially one in the form of an artifact to be produced, certain software and hardware requirements must be met, as they constitute some of the major components needed to develop a whole system. Thus, in this study, some software application tools and hardware components were used as discussed further in section. The software applications and tools used to develop the fail-safe system are categorized into three distinct group based on the development process of the system. During the test client development stage, the NetBeans IDE V8.0.2 was used together with the JavaFx Scene Builder for the design and customization of the user interface of the BankApp. This is because the software applications provide flexibility and customization within a GUI based on object-oriented programing. At the design and development of the datacenter, the Linux (Fedora) operating system was used to configure and mount the Primary (Master) and Secondary (Slave) servers (Figure 10) using the feature of virtualization contained in the VMware Workstation (V12 Pro) to allow real-time replication of information between the two servers as well as provide the administrator a wider range of options in terms of system configuration and controls. Additionally, while the ERD was designed using the MySQL workbench (V6.2) for the management of the test client's database, Java programing language was used for the



```
[root@localhost abdul]# vi /etc/my.cnf
[root@localhost abdul]# systemctl restart mysqld
Job for mysqld.service failed because the control process exited with error code
.
See "systemctl status mysqld.service" and "journalctl -xe" for details.
[root@localhost abdul]#
[root@localhost abdul]# mysql -u root -p < /root/dbdump.db
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/lib
/mysql/mysql.sock' (2)
[root@localhost abdul]# systemctl restart mysqld
Job for mysqld.service failed because the control process exited with error code
.
See "systemctl status mysqld.service" and "journalctl -xe" for details.
[root@localhost abdul]# vi /etc/my.cnf
[root@localhost abdul]# systemctl restart mysqld
[root@localhost abdul]# mysql -u root -p < /root/dbdump.db
Enter password:
[root@localhost abdul]# systemctl restart mysqld
[root@localhost abdul]# █
```

Figure 10.
The server configuration process

overall design of the fail-safe system because it is object-oriented and supports the development of a more efficient, secure, reliable and user-friendly system.

Although the fail-safe system can be run on virtually every type of system with common minimum requirements, it is best that the system is run on the following minimum requirements in Table I to achieve optimal efficiency and speed considering the environment that the system would be used.

8. Results from evaluation and testing

The main purpose of conducting this evaluation is to determine the resulting effect, i.e. to test whether the designed fail-safe system is capable of maintaining the availability of the BankApp through real-time replication of transmitted of stored information. To determine this, three tests have to be run including a test to initially show that both servers are working normally, another to show the presence of an attack on the Master server as well as a test to show that the Slave (backup) server not only automatically picks up to continue with normal system operations but also enables replication continuity onto its sub-servers in case of similar re-occurrence of circumstance. In addition, it is important to note that all these sets of activities should occur so fast (seamlessly) without the end-user interrupted from any of his activities before or while the failed Master server is attempted to be restored back to normal operations.

8.1 Test I

The first evaluation is used to determine whether or not both the Master and Slave servers are working normally at the same time. Plates 1 and 2 presents a screenshot of both the Master and Slave servers up and running on the virtual machine at the same time with their individual status labeled and waiting to take in inputs through the BankApp. Plate 1 indicates the port number and the IP address that connect the primary and the secondary server together. The Binary Log (Binlog).

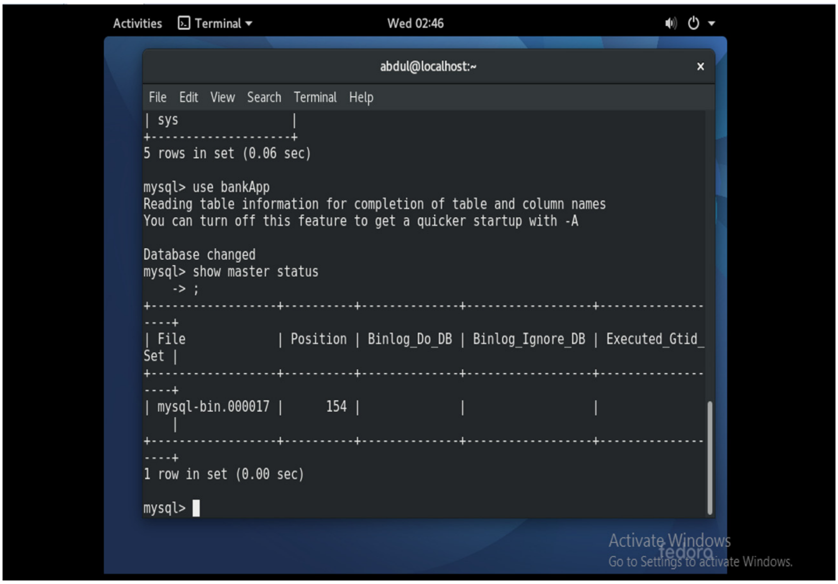
Plate 2 shows that the Slave is running and ready to listen the Master Server to get update as data are been entered from the BankApp. The port number when compare the master server indicate that Plate 2 is actually the slave server

8.2 Test II

Here, the present of a distributed denial of service (DDoS) attack is tested on the servers to determine what occurs at the server end between the Master and Slave servers in terms of the availability of the system. Considering that a DDOS attack can be executed either through the use of malicious codes or by the physical termination of infrastructure connection, the latter is used for this particular test. At this point, after the Master server becomes unavailable completely, (as in the case of a physical DDoS attack), the primary server is not able to continue with normal operations and replication. As shown in Plate 3,

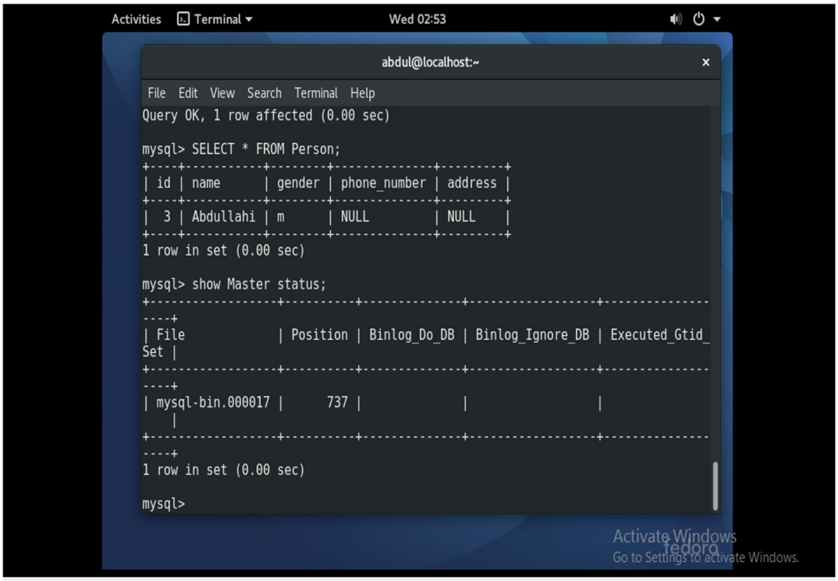
Components	Specification
Operating system	Windows 8, 9 or 10 (any Edition)
Visual Machine Tools and OS	Fedora Linus OS and VMware Workstation
Processor	Intel Core Dual Processor
Memory	8 BG Ram
Hard disk drive	512 GB
System architecture	64-Bit OS

Table I. Hardware requirement specification


Activities
Terminal
Wed 02:46
abdul@localhost:~
File Edit View Search Terminal Help
Activate Windows
Go to Settings to activate Windows.
fedora

```
| sys                |
+--------------------+
5 rows in set (0.06 sec)

mysql> use bankApp
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show master status
    -> ;
+------------------+----------+--------------+------------------+--------------
----+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_
Set |
+------------------+----------+--------------+------------------+--------------
----+
| mysql-bin.000017 |      154 |              |                  |
    |
+------------------+----------+--------------+------------------+--------------
----+
1 row in set (0.00 sec)

mysql> 
```

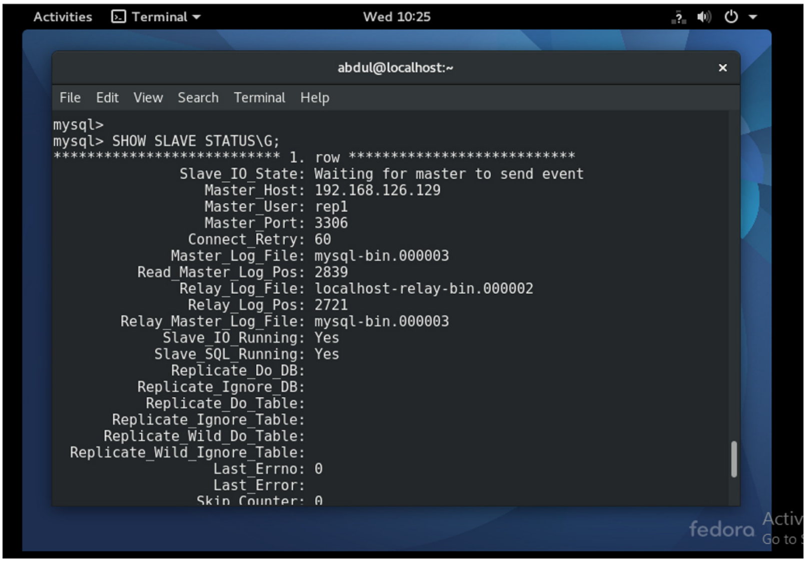
Plate 1.
The screenshot of a normal operation of a master server


Activities
Terminal
Wed 02:53
abdul@localhost:~
File Edit View Search Terminal Help
Activate Windows
Go to Settings to activate Windows.
fedora

```
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Person;
+----+-----------+--------+--------------+---------+
| id | name      | gender | phone_number | address |
+----+-----------+--------+--------------+---------+
|  3 | Abdullahi | m      | NULL         | NULL    |
+----+-----------+--------+--------------+---------+
1 row in set (0.00 sec)

mysql> show Master status;
+------------------+----------+--------------+------------------+--------------
----+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_
Set |
+------------------+----------+--------------+------------------+--------------
----+
| mysql-bin.000017 |      737 |              |                  |
    |
+------------------+----------+--------------+------------------+--------------
----+
1 row in set (0.00 sec)

mysql>
```

Plate 2.
The screenshot of a normal operation of a slave server

slave server has automatically and immediately taken back full control of the system from the failed server within a short time period (in seconds), the slave takes over and become the master server and the normal operation continue without Bank customer and staff noticing the failure.



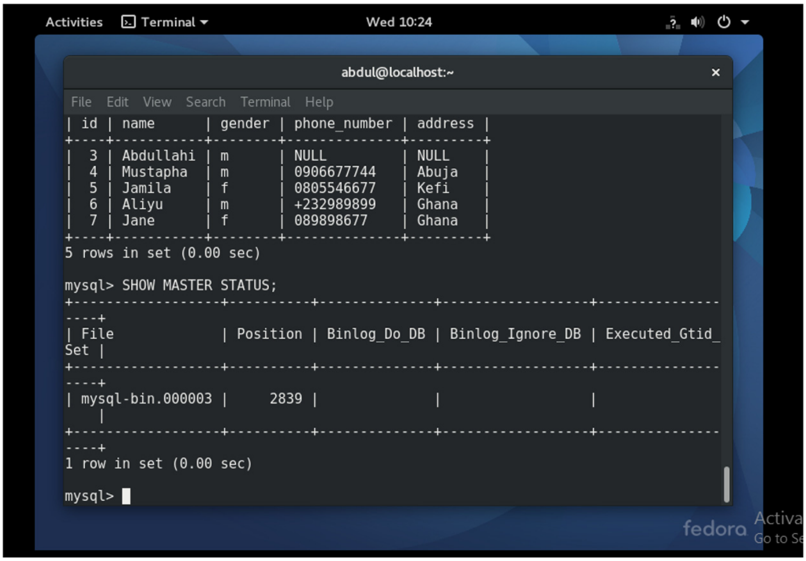
```
mysql>
mysql> SHOW SLAVE STATUS\G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 192.168.126.129
                  Master_User: repl
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000003
          Read_Master_Log_Pos: 2839
               Relay_Log_File: localhost-relay-bin.000002
                Relay_Log_Pos: 2721
        Relay_Master_Log_File: mysql-bin.000003
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
```

Plate 3.
The slaver (new master)

Plate 3 indicates that the slaver becomes the master server and takes in inputs from the BankApp while the earlier master is attacked.

8.3 Test III

At this point, since the Master server has failed and the slave server (new master) continued with the running of the system, Plate 4 presents a screenshot showing the



```
| id | name      | gender | phone_number | address |
+----+-----------+--------+--------------+---------+
|  3 | Abdullahi | m      | NULL         | NULL    |
|  4 | Mustapha  | m      | 0906677744   | Abuja   |
|  5 | Jamila    | f      | 0805546677   | Kefi    |
|  6 | Aliyu     | m      | +232989899   | Ghana   |
|  7 | Jane      | f      | 089898677    | Ghana   |
+----+-----------+--------+--------------+---------+
5 rows in set (0.00 sec)

mysql> SHOW MASTER STATUS;
+------------------+----------+--------------+------------------+---------------
----+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_
Set |
+------------------+----------+--------------+------------------+---------------
----+
| mysql-bin.000003 |     2839 |              |                  |
    |
+------------------+----------+--------------+------------------+---------------
----+
1 row in set (0.00 sec)

mysql> 
```

Plate 4.
Master server get information from the slave

extension of replication activities by the Slave server onto its sub-servers to ensure that real-time backup or replication of information is continued immediately. Furthermore, when the old Master server is eventually restored (either manually or automatically), the Slave (new master) server updates the old master all the data entries that occurred while it is attacked, the new master become the master because it has less port number as can be seen in Plates 3 and 4.

Plate 4 shows that the attacked server gets updated from the slave server when it comes up. This server seize to be the master, thus change status to a slave and continue to get updates from the master until one of the server is attacked and the process continues.

9. Conclusion

Information systems continue to face hard and threatening challenges of cyber-attacks, vulnerabilities and risks in a time where organizations, governments and even individuals highly depend on these systems for the execution of their various daily activities. In any case, where any of these entities is breached security-wise, the confidentiality, integrity and availability of information in transit or storage may likely be in jeopardy. Hence, valuable resources, market and business confidence amongst other factors could be lost.

This paper has presented a designed and developed fail-safe system for the cyber-environment using the instance of the network interdependency concept to illustrate how key sectors and organizations of a Nation depend on each other's survival for their operations as well as how the system can be implemented to prevent or mitigate against unavailability, integrity and confidentiality within the cyber environment in the case of a system security breach. Furthermore, this system has been tested and evaluated through simulation to show how it effectively functions to solve the problem in question. The cyberspace and concept of network interdependency, the problem statement and the aims and objectives of the study and been tackled. The paper presented the design and approach to model development with novel tools and techniques used to develop the fail-safe system. This paper also gives an overview of system model, the design and development process, requirement specifications, evaluation and testing of the system while briefly recapturing the overall data and model flow process. Following the conclusions presented here, the developed fail-safe system with MySQL replication technique is highly reliable to organizations, which would help mitigate the impact of resource loss in an event of a system security breach while improving business and market confidence outside the organization.

The role and benefits of this paper to Crowd Science in areas of cyber physical/information systems in the instance of the network interdependency concept, involves the mitigation of the fail-safe system which was presented in all aspect of the paper. The system can be implemented within the cyber-environment to improve the security of information systems for organizations. With reference to network interdependency, this research can be used to mitigate or prevent against the impacts of cyber-security breach, considering that no cyber system is fail proof. Critical information transmitted or stored on an organization's server would largely benefit from this system in terms of efficiency and full-on availability of service, which in turn, increases customer and market confidence as well as save the resources of an organization.

The contribution to knowledge here is that, while there are fail-safe mechanism such as system recovery used in Microsoft office, system backup used by many organization, non-have considered the system in an interdependent environment (cyber-environment) where critical sector such as telecommunication, banking/finance and financial activities by banks customer completely depends on one another, this paper has provided a model that addressed a fail-safe in an interdependent environment. In summary, the achievements here include:

- to provide fail-safe in interdependent system environment;
- safer to implement a fail-safe mechanism at data-center to avoid duplication in a multiple environment; and
- continuous provision of service(s) to customers and staff on the event of DDoS attacks.

Future recommendations includes application of a fail-safe method at the data-center, a master–master architecture should be used to implement a fail-safe technique and considering the size and scale of the cyberspace including the high number of users and devices involved, certain limitations exist in terms of delivering a wider or more complex study. In the future, other methods can be explored such as the use of clustering technique at the data-center instead of replication. In addition, a more robust system can be developed to function more effectively such as a fail-safe system that also consists of more components like monitoring, detection and prevention unit.

References

Arshad, J. and Matt, L. (2009), *Issue on the Design of Efficient Fail-Safe Fault Tolerance*, Vol. 10, IEEE, Piscataway, NJ, pp. 23-29.

Bishop, M. (2003), *Computer Security: Art and Science*, Vol. 12, Addison Wesley Professional, Westford, MA, pp. 23-34.

CERT UK and GCHQ (2012), *Common Cyber Attacks: Reducing the Impact*, Vol. 44, CERT-UK, pp. 24-28.

Chakraborty, A. (2009), *Fault Tolerant Fail System for Railway Signalling*, Vol. 12, WCECS, San Francisco, pp. 23-34.

FFIEC (2015), *Cybersecurity Assessment Tools*, UK FFIEC, Vol 12, pp. 23-34.

Gollmann, D. (2005a), "Why trust is bad for security", *Proceeding for the International Workshop on Security and Trust Management*, Vol. 12, *ENTCS*, pp. 23-34.

Gollmann, D. (2005b), *Computer Security*, Vol. 45, John Wiley and Son, Hoboken, NJ, pp. 34-56.

Gollmann, D. (2005c), "Why trust is bad for security", *Proceeding for the International Workshop on Security and Trust Management, ENTCS*, Vol. 45, pp. 34-56.

Gollmann, D. (2006), *Security Engineering*, Vol. 45, 2nd ed., Wiley, Hoboken, NJ, pp. 34-66.

Greene, L. (2012), *Fail Safe vs Fail Secure: When and Where*?, Vol. 3, Door and Hardware Institute, Chantilly, VA, pp. 45-67.

Guiho, G. and Hennehert, C. (2000), "Software validation in ICSE", *Proceeding of the 12th International Conference on Software Engineering, IEEE Computer Press*, Vol. 56, pp. 67-76.

Ida, M., Abdul, G.A., Sonny, Z., Sigit, P. and Wigati, J. (2013), *Data Breach on the Critical Information Infrastructures: Lessons from the Wikileaks*, Vol. 6, IEEE, pp. 34-45.

Jack, F.A. (2011), "From fail-safe to safe-to-fail: sustainability and resilience in the new urban world: landscape architecture", *Regional Planning Studio and Student Research and Creation, Activity*, Vol. 8, pp. 23-34.

Ken, H. (2003), "Vulnerability and vulnerability scanning. A white paper to help small and medium scale business understand the issues and some proposals to help resolves issues", SAN Institute.

Larman, C. (2005), *Applying UML and Pattern: An Introduction to Object Oriented Analysis and Design Iteration Development*, Prentice Hall PTR, Upper Saddle River, NJ.

Leidigh, C. (2005), *Fundamental Principles of Network Security*, American Power Conversion (APC), RI.

Locked, M. (2014), Lockheed martin corporation, Annual Report.

Mbanaso, U.M., Chukwudebe, G.A. and Atimati, E.E. (2015), *A Critical Assessment of Nigeria's Presence on the Cyberspace*, IEEE, p. 11.

Sajal, S., Sudip, S., Kajal, S. and Soumalya, G. (2015), *Cyber Security Password Policy for Industrial Control Networks*, Vol. 6, IEEE.

SANS (2002), "An overview of threat and risk assessment", Developing a security-awareness culture: improving security decision making, SANS, Semantec (2012), Internet security report, Semantec corporation Volume 17, SANS Institute.

Semantec (2011), *Advanced Persistent Threat: A Symantec Perspective, Preparing the Right Defence for the New Threats Landscape*, Semantec Corporation, Mountain View, CA.

Steve, F. (2015), "Oracle database administrative guide, 11g release 2 oracle and/or its application".

Willian, F.C. (2015), *Cybersecurity Kill Chain*, Vol. 10, ISACA.

Zahri, Y. and Syahrul, H.S. (2010), *Safeguarding Malaysia's Critical National Information Infrastructure (CNII) against Cyber Terrorism: Towards Development of a Policy Framework*, Vol. 7, IEEE.

Further reading

Yinghua, M., Yutang, Z., Zhongcheng, L., Cheng, Y. and Yuqi, P. (1994), *Behavioral Design and Prototyping of a Fail-Safe System*, Vol. 4, IEEE.

Corresponding author
Fatima M. Isiaka can be contacted at: fatima.isiaka@outlook.com