Securing a Bring Your Own Application cloud environment using digital forensics

Duncan A.Litunya
Faculty of Information Technology (FIT)
Strathmore University

Securing a Bring Your Own Application Cloud Environment Using Digital Forensics

By

Duncan Oyando Akhonya Litunya

089591(Litunya, 2018)

Submitted to the Faculty of Information Technology in partial fulfillment of the requirements for the award of Masters of Science in Information System Security at Strathmore University

Faculty of Information Technology

Strathmore University

Nairobi, Kenya

May 2018

Declaration and Approval

Declaration:

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the research contains no material previously published or written by another person except where due reference is made.

Student Name: Duncan Oyando Akhonya Litunya

Sign: _____

Date: 16th April 2018

Approval:

The dissertation of Duncan Litunya was reviewed and approved for examination by the following:

Supervisor's Name: Dr. Bernard Shibwabo Kasamani

Sign: _____

Date: _____

Abstract

The use of cloud applications introduces new challenges to information systems Security. The idea of applications accessible from multiple devices and hosted or provided by third party organizations brings new complications to IT security. In situations where organizations are embracing Bring Your Own Applications (BYOA) and where they allow use of free to public cloud applications within their networks, it is important for IT Security experts to consider how to secure their BYOA environments and also monitor how these applications are used and the flow of information. The aim of this research is to develop a digital forensics based solution for securing BYOA cloud environment. This solution can be used to improve security in an organisation implementing BYOA. The research focuses on free to public cloud applications, whereby security challenges are identified and security measures proposed. The security measures are enforced through the development of a customized solution. The solution has been developed using rapid application development (RAD) system development methodology. Using Geany editor and Python programming language, the prototype developed relies on digital forensics artefacts to gather information about the usage of BYOAs. The solution captures digital forensics artefacts and stores them into a database as logs of the activity on Google Drive application. The solution demonstrates how digital forensics artefacts can be used to enhance security in a BYOA environment.

Keywords: Digital forensics, IT Security, BYOA, SaaS Forensics, Cloud Forensics in SaaS, Security Intelligence in SaaS, Google Docs Forensics.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background

Companies are embracing cloud computing and other models spawned from cloud computing like bring your own application (BYOA). There are many factors behind this; employee productivity, staff mobility, costs – it is cheaper than enterprise solutions especially for smaller organizations and also considering licensing and other related software costs (Rouse, 2016; Comcast Business View, 2016).

However, the risks of adopting this model of computing are also big. There is almost complete loss of control of applications and to some extent data. How can an organization get back some of the control (Green, 2015; Patel, 2014)? In Africa, especially for small and medium enterprises (SMEs), BYOA is very attractive but the same cannot be said in organizations where security is of high importance. Implementing BYOA involves balancing security requirement, budgets available and the risk appetite of the organization.

Cloud computing is a model that provides ubiquitous, convenient, on demand network access to a shared pool of computing resources enabling organizations to increase computing capacity or capabilities without heavily investing in capital expenditure (Grance, 2009). Cloud computing has transformed how IT services are managed, accessed and delivered. There are various types of cloud computing delivery models; IaaS – infrastructure as a service, PaaS – platform as a service and SaaS – software as a service (Grance, 2009).

Bring your own application, BYOA sometimes also referred to a build your own application is a growing trend that allows employees to use their preferred applications for work purposes (Green, 2015). If implemented as a strategy it has a very low cost barrier and together with bring your own device they form the cornerstone of bring your own everything strategy (Akpose, 2014). BYOA in respect to this research specifically looks at those delivered as SaaS model and as a free service to the public. The companies ‘selling’ these applications as a service make money by getting more people to use the internet and also through selling advertisements.

There are a wide variety of free to public cloud applications, popularly referred to as consumer versions/ applications (Akpose, 2014). These applications are being adopted as BYOA, sometimes even without the knowledge of the organization itself, in such situations they are said to operate within 'shadow IT' of the organization. Employees use these consumer applications to access enterprise systems and also store organizational data (Akpose, 2014). Some of these cloud applications do not need installation into the device, they can be accessed via web browsers.

However, BYOA poses security challenges. Consumer adoption of smartphones has encouraged the culture of "apps" which is a popular moniker used to refer to applications. Mobile phones have morphed into mini-computers, now applications are providing the same experience in computers and in mobile devices (Mordhorst, 2014). This is driven by users demand to have similar experience on computers and mobile devices. This in turn makes users more comfortable. Enterprise systems have responded by breaking down 'big systems' into modules and allowing users to choose what they want or feel comfortable with.

1.2 Problem Statement

BYOA implementation poses security threat especially through information leakage (Walters, 2013). Information can be moved from one point to another, or other crimes or violations can occur in a BYOA environment. When incidents do happen, an organisation will only have two sources of information where they have complete control; their device and their network.

In Africa, organizations are embracing BYOA for various reasons. With internet penetration increasing and the average Internet speeds also increasing more people are using mobile devices and home computers to access enterprise applications. BYOA model also means that not everything is under the absolute control of the IT department of the organization (Rouse, 2016). A third party; the cloud provider is added into the picture (Rouse, 2016). For IT Security experts and Digital Forensics practitioners, examiners and researchers this is a new challenge.

When implementing BYOA, the area where the organization can fully have total control is only the device (if it belongs to the organization) and their network (when it is used to access internet). It is not entirely possible to get cooperation of the cloud providers, except in criminal cases, in many of the incidents leading to a forensics investigation or security analysis. Even in cases where

the cloud provider can be involved the process, it is not entirely clear (in law or practice). The response time can also take long, whereas time is of essence in any incident involving cloud computing. It is better to assume that when an incident emerges, the organization can only use the infrastructure under its control. This is where they can exercise absolute control to find answers as to what exactly transpired. Apart from policies, the organization needs to have capacity to determine where their data is (to an extent) and what is happening in their network with applications currently in use.

1.3 Aim

The aim of this research is to develop a digital forensics based solution for securing BYOA cloud environment. This solution can be used to improve security in an organisation implementing BYOA.

1.4 Specific Objectives

i. To investigate the factors relating to security in BYOA cloud applications.
ii. To analyse available tools and techniques that provide security for BYOA cloud applications.
iii. To develop a digital forensics solution to enhance security in BYOA cloud applications.
iv. To validate the developed solution.

1.5 Research Questions

i. How can issues of the security in BYOA cloud applications be investigated?
ii. What are the security concerns when implementing BYOA?
iii. How can a solution, using digital forensics artefacts, be developed to secure BYOA cloud applications?
iv. How can a solution of BYOA security be tested and validated?

1.6 Scope and Limitations

The research and solution is only specific to Google Drive. It focuses on access via installed clients using web access. It was limited to windows based computers. The research looks into possible

places where information can be stored and retrieved. The internet browser used for this purpose is Google Chrome for purposes of interaction with Google Docs which is part of Google Drive. The prototype solution developed aids in digital forensics artefacts acquisition and also logging activities of the cloud applications. The research involves participants from the following countries; Uganda, Tanzania, South Africa, Ethiopia and Zimbabwe. These participants offer their expert opinion as IT managers.

1.7 Justification

As organisations seek to implement BYOA, it is important to know what can be achieved and what gaps remain in terms of security and digital forensics. This research is meant to provide information about the current situation in terms of digital forensics and security for BYOA and also provide solutions where there are security and digital forensics gaps. Organizations implementing BYOA can learn from this and also use this as basis to build custom made tools of their own. BYOA requires new policies and tools deployed to reduce risks and keep up with growing user expectations. Balancing accessibility and security has proven to be a security challenge (Seth Early, 2014).

The biggest issue preventing companies from implementing BYOA is data security (Log Me In, 2013), for small and medium size enterprises this is the bleak reality. Larger organisation are able to leverage on their huge budgets to put measures in place. Smaller organisation that have IT security and IT operations rolled into one require innovative thinking and low cost solutions. The real challenge for IT is how best to protect and govern data particularly when it is being exposed to the cloud through cloud storage applications, and documents are created and shared through productivity suites (Walters, 2013). There is the risk of data leakage and corporate reputation management (Walters, 2013).

There is a problem of ownership of documents stored in these applications, while the documents/ data belong to the organisation, the employee holds the key. In the wake of Hays Vs Ions case, employment lawyers are encouraging organisations to update their employment contracts to cater for the surrender or deletion of client data stored in personal accounts in cloud applications

(Walters, 2013). IT needs to provide information that BYOA is governed, monitored and audited (Walters, 2013).

Chapter 2: Literature Review

2.1 Bring Your Own Application

In 2012, a survey was done involving around 3,000 IT managers in 29 different countries. The purpose was to discover the extent of 'shadow IT'. More than 80% of the managers acknowledged that employees had 'procured' cloud based applications without the involvement of IT. More than 70 % also discovered instances of cloud based services being used without prior involvement of IT (Walters, 2013). Social applications are the most used BYOAs, others are; cloud storage/ sync, collaboration, remote access and also productivity tools.

In most scenarios, organisations only respond to these applications when they have gained critical mass acceptance and it is not mainly for security purposes (Log Me In, 2013). Organization respond through endorsements for purposes of uniformity and standardization. Employees will continue to use their favourite applications and these applications often come from different and sometimes obscure software publishers. Evidence shows that many applications misappropriate data or access data that they should not (Log Me In, 2013).

2.2 The Bring Your Own Application Debate

Despite security concerns, organisations are implementing BYOA and some have it within their shadow IT, operating without IT input but not blocked by IT. This is because there are some benefits of using BYOA. BYOA offers a boost to employee productivity, they use tools they are accustomed to, improving performance and work quality (Log Me In, 2013). Employees also do not need to wait for IT to develop a solution for a particular business problem, they can think and use a solution that works for them, and they eventually fill in gaps that are within their current setup. Even in heavily regulated and controlled IT environment, the different applications can also be interfaced allowing an integrated setup.

Most applications are also mobile based allowing employees to work on the move. BYOA is also cheap, keeping the security aspect out of this argument, BYOA (free to public cloud applications) is a very low cost solution. No license fee and no consultancies required for installation, it is similar to plug and play devices – download, register and use. Flexibility is also another key advantage, BYOA allows employees to use the right tools for the right job, enterprise IT applications are known to be rigid (Log Me In, 2013).

However, consumer applications are not designed for the corporate world. Although some software publishers are aware of this and are developing for the corporate world, the security concerns for IT will remain. Some corporates are even adopting innovative ways to deal with BYOA – like corporate application stores. Data security in the cloud (Log Me In, 2013) is the top concern for BYOA, and cloud or sync storage applications top the list of data security concerns both from IT and users perspective. BYOA means loss of control for the IT department. However, IT can control security by enforcing security measures on devices and data within the corporate network. BYOA also introduces mobility from one device to another and cloud storage where IT cannot exercise control.

Integration is not easy within the BYOA environment, integration involves access controls, information sharing and standard security policies. Although achievable to some extent, it is not as perfect as enterprise integrated IT systems and will also involve some restrictions to applications that can be integrated. However, it is important to note that the trend in mobile business applications is access to real-time enterprise data, BYOA will continue to grow.

2.3 Implementing Bring Your Own Application

When implementing BYOA, IT has to recognise the objective is to bridge the gap between IT technical and the business processes that employees undertake on day to day basis. Enterprise IT concentrates so much on the bigger picture that they created a gap that is being exploited via consumerization of IT. If the same approach is taken for BYOA as enterprise IT, it will end up in the same gridlock. BYOA governance involves balancing employees' flexibility and freedom to solve problems while implementing a set of guidelines that work to create accountability (Moss, 2015).

It is important to have employee engagement and collaboration when defining the BYOA road map. It is important that employees are aware of what enterprise IT has to offer vis a vis the BYOA applications available (Moss, 2015). Clear guidelines have to be developed, defining what can, and cannot be done, detailed instructions/ wikis and an IT approved list of applications is also important. The list should be regularly updated to prevent it from being irrelevant. Also a process of listing problems that require IT oriented solutions can be maintained to encourage staff to look for solutions in the 'application world'.

Where possible the organisation should implement a private application stores. Though for SMEs this a huge challenge and is opposite to the entire purpose of implementing BYOA, this approach is for organisations that have high security compliance requirements and enough budgets to implement. A different approach is utilising only approved public application stores where there is some form of control on the applications available (Moss, 2015). It is important to have management buy-in and governance guidelines. The BYOA management procedures need to be approved at the highest level with consequences of infractions enforced and endorsed by top management (Moss, 2015).

Data protection is critical, most applications will transfer, store and exchange data in the cloud. It is important to recognise the risks especially in highly regulated industries. Various applications have add-ons that allow corporate data to leak into other areas; example Outlook social connector – corporate address book accessible from Facebook (Microsoft, 2014). There are a lot of weak points when dealing with BYOA and the adoption of consumer cloud applications. Protecting the devices themselves does not guarantee prevention of data leakage. It is very important for companies to actively protect their online assets. They should know the risks associated with BYOA applications the information the applications store on the cloud and on which devices. An organisation should be able to discover, analyse and control these applications, understand the associated risks and enforce policies.

Within the BYOA, the protection mechanism is mainly concentrated on 'your side of the fence'. The devices and the network become very important in securing the environment or at least implementing measures to mitigate the risks identified.

2.4 Security in Bring Your Own Application Environments

In the world of BYOA it is almost impossible to draw the line between personal and professional life. The line between work hours and personal time became almost indistinguishable during the internet boom. Smart phones and tablets make it impossible to distinguish between personal and work related computing device. Employees are increasingly mobile, for such employees there is a mixture of personal and work related applications installed. Some application may cross the boundaries and service both. Employees also work from home, a home computer can be used for both office and personal functions. This mixture of devices will steer users to using popular BYOAs like Google Drive which eventually make their work easier. These paradigms raise a

dilemma: there is no longer the traditional IT perimeter that guards the enterprise assets. There is lack of separation of personal and private information from corporate intellectual property and some data in storage on a device can be a liability for the employer (Li & Clark, 2015).

BYOA cloud moves the enterprise from locally hosted solutions to cloud hosted collaborative model. This means access to information through web browsers and sync clients, the traditional approach of IT security is inadequate. IT needs to monitor, audit and govern the use of BYOA, yet no monitoring or controlling tool is completely adequate (Bennett, 2016). Organization have developed various workarounds to achieve this. Some strategies include Wi-Fi 'guest networks' which are exclusively used to access BYOA and monitor while preventing simultaneous access to corporate data, mobile management tools which are used to manage and control mobile devices, and some network packet monitoring tools.

Another approach is using cloud access security brokers (CASBs), they have been developed to provide policy enforcement. They are the gate keepers and are situated between the cloud service users and the providers. They offer the following services, authentication, single sign-on, credential mapping, profiling, encryption, comprehensive logging and alerting (Gartner, 2017). While offering comprehensive security, this is another solution for organisation with big pockets. Apart from that it brings in another layer of complication, CASBs becomes another cloud provider to deal with when reviewing incidents or there is need to analyse data.

2.5 Digital Forensics and Security

Digital forensics involves obtaining valid evidence of an event or cyber security incident that can be a violation of policy, system or a crime. An event is any observable occurrence in an application or network, it can be a user opening a file or sending an email. Events can be normal or adverse; adverse events have negative consequences. These consequences can be system crashes, unauthorised file access, data destruction or execution of malware. Cyber security incidents are a violation or imminent threat of violation of computer security policies, acceptable use policies or standard security practices (NIST, 2012).

Incident response involves detection and containment of cyber security incidents. The focus is on quick remediation and return to normal business. Incident response follows a very structured process, digital forensics becomes very important during the investigation of incidents, especially

involving data collection and analysis (Freiling, 2007). Digital forensics involves obtaining, analysing and presenting evidence. For the purpose of securing BYOA, emphasis is placed on obtaining evidence. Going beyond incident management and the objective being overall security, digital forensics methods can be used to collect evidence of events, adverse events and cyber security incidents. Further analysis of this evidence can be used to also come up with improved security measures or different tactical approaches to security.

Retrospective analysis works with data that was collected in the past. From this data various observations can be made, detailed information can be obtained of incidents and additional events related to that incident. This information can used to draw a conclusion (Li & Clark, 2015). For such a system to facilitate retrospective analysis the following functionalities should be taken into account:

Data acquisition: properly defined parameters of occurrences of events that are deemed important within the BYOA The acquired data should contain enough information to elaborate on the event. Example: file name, file owner and time stamps.

Data organisation: Acquired data should be organised in such a way that it is easy to manage, process and retrieve. It should be possible to query the data, get rapid responses that match the criteria. Storage of this data is also important.

To attempt to create as secure environment in BYOA should involve some aspect of security intelligence. This must be supported by a robust data collection mechanism.

2.6 BYOA: Google Docs and Google Drive

Google Docs is a free Web-based application in which different types of documents can be created, edited and stored online. Files can be accessed from any computer with an Internet connection and a full-featured Web browser. Google Drive is cloud storage that offers free storage up to 5 gigabytes (TechTarget, 2017). Google Drive is used to refer to both Google Docs and Google Drive. Considering that this environment lacks centralised control, file activity cannot be easily governed, controlled or traced. The user has absolute control over the features of cloud storage. Limited cooperation is expected from the cloud provider, Google, who provide this service as a free to public offering when investigating incidents or events.

Most evidence of usage is based on traces left by the client application or web activity. In a normal windows desktop most cloud storage applications will have logs and other information stored. Google drive uses SQLite3 databases to store information and activity of the client application. It is best to use digital forensics methods to gather information when evaluating security incidents and events. It is also important to use digital forensics techniques to unpack what this information really is and draw appropriate conclusions based on solid evidence. It is important to note the application revolution brings total power to the end user and robs enterprise IT of its traditional duties and powers. Enterprise IT risks being negated to the occasional helpdesk call and will lack real power to control risk and security incidents (Garrison, 2010).

BYOA presents a whole new set of risks, challenges and opportunities, and security experts are the most affected (Garrison, 2010). IT no longer directly manages servers, applications, networks and devices; laptops, desktops and phones, they are still managed to some extent but this control is not as it was 10 years ago. Employees are now more knowledgeable to use and install (some do not need installation) applications without IT help. IT needs a solution to help log and translate the digital forensics artefacts about the usage of Google Drive.

2.7 Conclusion

BYOA complicates life for IT, security being the biggest challenge. Whereas internal applications typically rely on firewalls, BYOA rely on secure passwords and encryption. Internal applications usually have monitoring and auditing through logging. Monitoring BYOA is a big part of implementing BYOA governance (Stuart, 2016). Large organisations have the upper hand in implementing new service gateways for enabling, securing and monitoring BYOA, SMEs on the other hand lack budgetary muscle and need alternative ways to monitor BYOA. Monitoring the BYOA applications, especially file synchronisation application can help IT in two ways; first it is a security issue and secondly it can help develop future cloud strategies based on how employees are using current BYOAs (Parizo, 2016). Security analytics and investigations are now a big part of information security, logging and monitoring are important elements that facilitate this. IT needs to be able to monitor and log activity of BYOA.

Chapter 3: Methodology

3.1 Introduction

This chapter describes the system development methodology used to achieve the goals for the research and development of the solution. The approach of rapid application development (RAD) using a qualitative-exploratory research was chosen to be the most appropriate as it is important to get the most accurate views of potential users. The realities of operations within an IT department must be captured and a solution relevant to these realities developed. The approach taken is team based and allows potential users of the system to participate or give input for the development and refinement of the solution.

3.2 Rapid Application Development

The system development methodology used in this research is Rapid Application Development (RAD). RAD is a development lifecycle designed to give much faster development and higher quality results than those achieved with the traditional software development lifecycle (Martin, 1990). It is an incremental model where a prototype is produced and improved in an iterative approach based on input from users and developers. RAD provides the abilities to quickly develop an application and to make modifications when needed.

A Solution that meets the major objectives of the project is developed quickly and modifications/ refinements are added to improve on features and functionalities. This allowed for a prototype to be developed quickly within the limited time constraints of this project. RAD reduced the traditional waterfall model into four steps. These four steps are a compressed version of the waterfall model and form a cycle of iterations. These four steps put more emphasis in analysis and design.

The key phases in RAD were:

System Analysis: The problem was defined as this stage; it was the initial process to gather the specific requirements of the system. This was the planning phase that determined the system scope.

System Design: The requirements were then analysed and transformed into logical then physical systems specifications and descriptions. Processes within the system were defined including all critical system components.

Development / Construction Stage: System code was generated as well as database descriptions. A prototype was built then tested and feedback was provided that was used to refine the prototype. The feedback and modification cycle continued until a final, acceptable version of the system emerged. The initial prototype had limited functionalities and was improved with feedback from the focus groups until a final acceptable product was developed.

Testing: The system was tested and functionalities evaluated. The intention was not only to identify errors but to identify weaknesses and areas of improvement. Weaknesses and improvements were then integrated in the next iteration to improve the prototype.

RAD drastically reduced the time required to develop the application, it also gave greater control over project to the developer. End user satisfaction level was high because of the continuous involvement through the feedback processes within the methodology. Reusability of prototypes saved on time.

Rapid Application Development (RAD) further builds on the concepts of joint application development - JAD by starting all development processes at once (Osborn, 1995). However, project deliverables reach the customer in stages in the order of importance to the business process (Gottesdiener, 1995). It is important to note that RAD required the use of disciplined highly skilled workers to ensure the team meets the advanced schedule of testing, construction, and prototyping (Gottesdiener, 1995). To help meet such deadlines, each project is broken down into a set of “chunks” or “time boxes” that are grouped according to business priority (Gottesdiener, 1995).

Advantages offered by RAD include the ability of project sponsors to interact with project prototypes, which enabled them to provide better feedback to the developers (Osborn, 1995). The ability to pre-empt problems before they become big issues helped to ensure that costs remain tightly controlled while still ensuring a quality product. Due to the frequent consensus needed, RAD was the best technique especially when the system requirements were not well defined (Osborn, 1995). For the purposes of this research, a focus group of five acted as project sponsors. Figure 3.1 shows that various steps undertaken to achieve an acceptable functional model, it shows that only last three steps are iterative.

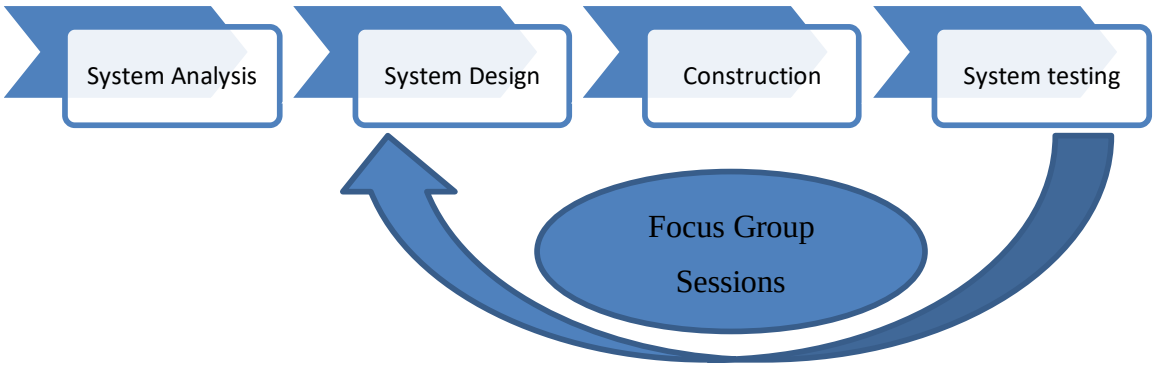


Figure 3.1 : Iterative Rapid Application Development

3.2.1 System Analysis

This was similar to the requirements planning phase, the focus group met to identify objectives of the application and to identify information requirements arising from those objectives. The team established a general understanding of the problem. This phase required intense involvement from all members; it was not just signing off on a proposal or document. The orientation in this phase was towards solving problems; what processes should be supported by the proposed application? A qualitative-exploratory research approach was undertaken. This was an unstructured method based on the teams input intended to provide answers to underlying issues, gain insights into the problem and discover new ideas of tackling the problem (Neelankavil, 2007). It helped in understanding the issues at hand thoroughly. It was useful in getting the real security issues for BYOA: Google Drive, looking at other measures that can improve the security and discover new ideas for implementing security. This was the platform for developing the solution.

The exploratory research was best considering the little knowledge available on this subject. It embarked on investigating and finding the real nature of the problem. The data collected was not through fixed–response questions, it allowed capturing of opinions and personal choices and even deviation from the subject line but not the research objectives. RAD and the qualitative-

exploratory research technique were complimentary, focus groups discussions were conducted during the initial system analysis stage and then on each subsequent iteration.

The team agreed on the business needs (relevance), project scope, constraints and system requirements (Shelly, 2009). The tool used to capture input and ideas from the team was mind mapping. Mind mapping helped to organize the conversation by aligning comments, requirements and ideas with the major thought branches in the conversations. A mind map is a hierarchical, visual organisation of information that shows relationships among pieces of a whole. This approach was used throughout all team meetings to aid in capturing improvements for the system. The team repetitively analysed in detail activities associated with the proposed system and proposed system functionalities until a satisfactory solution was produced.

Meeting transcripts were maintained for all meetings. These transcripts were organised into proper notes to produce a general understanding of the problem, a consensus on the objective and identified functionalities of the system.

The focus group consisted of five IT managers who work for small companies ranging from 10 – 20 users. These companies have Google Docs and Google Drive operating within their shadow IT. The team was used to refine the solutions' functionalities by providing personal, technical and expert opinion. This sample (focus group) did not employ the rules of probability sampling, nor claim representativeness but was best suited for qualitative-exploratory analysis. It was a non-probability purposive sample. The team involved in the research and development of the solution comprised of IT managers with experience and problems relating to BYOA: Google Docs.

Data and feedback was collected through workshop sessions conducted via Skype conferencing. Due to the geographic locations of each participant of the team, it was not possible to bring them all into one location due to time constraints and costs.

The team interactions allowed all members to make connection to various issues under discussion and provide individual perspectives.

Key elements;

i. Participants: 5 IT Managers.
ii. Environment: Virtual/ Skype video conferencing.

iii. Moderator: System Developer.

iv. Analysis and reporting: Minutes and mind mapping.

The moderator/ System developer also acted as the note taker. The moderator asked probing questions using an interview guide. Responses included opinions, suggestions and experiences that lead the discussion into a new direction provided it stayed relevant to the research (Sherraden, 2001).

The notes from the focus group meeting were organised and expounded; raw notes (meeting transcripts) and transformed into well-organized notes. The notes were ordered according to the research questions. Unexpected topics were also organised and a structured labelling of all topics was implemented. The research involved a lot of textual data and analysis was done manually.

3.2.2 System Design

This was a design-and-refine phase, looking into all system processes, outputs and inputs. The logical design was developed and then turned into a physical design. The physical design is the detailed description of what was needed to solve the original problem; inputs, outputs, databases and process specifications.

The data collected from system analysis was derived into system processes and functionalities, and further visually analysed using unified modelling language (UML). The data analysis was the basis of the system design in the system development methodology. UML was used to model the physical design (the structure and behaviour of the system). Structural and behavioural diagrams were produced from the system specification and functionalities determined in the team discussion sessions. The models were refined responding to actual inputs from the team in the iterative approach. This process was continuous and interactive, it allowed the developer to understand, modify and eventually produce a working model of the system (Shelly, 2009).

An outline system design was completed at this stage. Interactions between functions and data was identified and modelled. The models were further discussed and refined. Inconsistencies were resolved and redundancies eliminated during the subsequent iterations. Open issues were documented until they were resolved. A database for the solution was also designed at this stage using the entity relationship approach.

3.2.3 System Implementation

This was the construction phase, which focused on program/ application development. The focus group was still involved and suggested changes as functionalities are developed (Shelly, 2009) through the iterations process. A prototype was developed that can operate at an acceptable level. The system then underwent continuous development with input from the focus group until a complete acceptable functional application was produced and the detailed definition of the design of each functionality was completed.

The designs were translated into code. Geany a text editor was used to write the python code. The workspace: computer and development platform were prepped; installation of the required software was done. The database was also implemented using MySQL. The 'dummy' accounts for Google Docs were opened. The coding was implemented to achieve all identified functionalities to produce the prototype. System development and system documentation was done concurrently.

There was also need to conduct literature research on the implementation approach, this involved best coding practises using python code and digital forensics using python. Research was conducted based on other researchers who have done similar projects or used python programming.

3.2.4 System Testing

System testing was done using the iterative approach of RAD. The development was evolutionary and the system was improved as prototypes were produced and reviewed by the focus group. Testing was integrated all through the development cycle, prototypes were tested during every iterative cycle.

The testing was in two forms:

Verification testing: the system was put through a series of tests to ensure that each component of each system and complete system, functioned according to the defined user requirements.

Functional testing: test data was generated and used to verify the functional capacity of the system. These tests covered all scenarios including failure paths and boundary cases.

The same focus group was used to conduct the tests. The live demonstrations were done on the building platform and the focus group followed virtually through Skype meetings. Each participant was allowed to run their tests where possible and verify the results. This session formed the basis of the next iteration where required.

3.3 System Validation

Once the team was convinced that they had a product that solves the problem, the software was complete. There was no need for more iterations. The software now required validation from a different perspective. The software was submitted to two different IT managers for evaluation. This process was an open ended discussion based on the following lines:

Does the solution offer enhanced security on BYOA: Google Drive?

Do the functions within the solution cover all aspects of monitoring and logging and if not what areas are missing?

3.4 Ethical Considerations

It was important to note that this application had the capacity to collect personal information and probably infringe of people's privacy. The solution developed was for research purposes and testing was strictly limited to a controlled environment. This application was not used in any situation without duly informing the people who were monitored or affected by its usage.

Chapter 4: System Analysis and Design

This chapter provides details of how the system analysis and design was undertaken. It provides details of stakeholders and their roles in the development of the application. It details how the system requirements are produced then logical and physical designs developed from them. It also details how analysis of the digital forensics artefacts was done to determine what artefacts are important to be incorporated into the system.

The system is developed using RAD, as part of the development methodology a focus group acts as the main stakeholders to drive the process. The focus group consists of:

i. Primary users – IT managers who have Google Drive running in shadow IT or implemented as BYOA. Their role is to act as the primary users of the system and provide expert opinion on functionalities.
ii. Developer & Moderator – The person who develops the solution and moderates the focus group discussions.

The focus group objective is to provide a satisfactory solution based on the described concept of improving security in BYOA environment.

4.1 System Analysis

4.1.1 System Requirements

This stage involved working with primary users to identify objectives of the application and the information requirements arising from the objectives. Focus group discussion sessions were held involving intense participation from all members. The main goal was to solve the business problem – improve security in a BYOA environment. The first step was important, it forms the basis of all subsequent processes that follow; design, development and testing. The Analysis stage is not repeated in future iterations. It is integrated into the testing phase and forms the basis of the next design stage in subsequent iterations. The systems obtains data from other applications. The information it derives and stores is not to be categorised on how and who can use it but rather it is a proof of concept that such information can be collected, stored and ‘decoded’ to be understood by any user. Mind maps are used to gather the system overview and functional requirements based on the discussions.

4.1.1.1 Forensics Artefacts Acquisition

When Google Docs is installed, the following artefacts are created on the default installation location which is c:\ users\ <username>\AppData\Local\Google\Drive\user_default ; three SQLite databases – snapshot, sync_config and uploader.

Snapshot.DB: This database contains seven tables. Within these tables all the information of actual history of synchronization between the local computer and the cloud.

Sync_config.DB: This database provides information about the users account email address, local root path and Google Drive version.

Google Drive incorporates Google Docs and offers web-based office suites applications such as Word documents that allows users to create and edit documents online while collaborating in real-time with other users. To access or work on Google Docs offline, an extension must be enabled in Google Chrome browser. Google Drive client installation maintains different profiles for each user 'C:\Users\<username>\AppData\Local\Google\Drive\user_default\'.

On installation of Google Docs, different keys and values are entered into the registry, these keys and values can be used to identify the Google Drive client version and user folder for synchronisation;

i. SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\Folders\
ii. SOFTWARE\Google\Drive
iii. NTUSER\Software\Microsoft\Windows\CurrentVersion\Run\GoogleDriveSync
iv. NTUSER\Software\Classes

During installation, configuration files are saved inside the installation folder in the user profile, the executable and libraries are stored in the bin sub-folder and prefetch files are created in windows prefetch folder. Some information can also be derived from RAM analysis, using string searches. A process analysis of googledrivesync.exe produces the results in Table 4.1;

Table 4.1: Process Analysis

00000000`004bcd6e
"VS_VERSION_INFO"

```
00000000`004bcdca  "StringFileInfo"
00000000`004bcdee  "040904B0"
00000000`004bce06  "CompanyName"
00000000`004bce20  "Google"
00000000`004bce36  "FileDescription"
00000000`004bce58  "Google Drive"
00000000`004bce7a  "FileVersion"
00000000`004bce94  "2.34.5075.1619"
00000000`004bceba  "LegalCopyright"
00000000`004bced8  "Google"
00000000`004bceee  "ProductName"
00000000`004bcf08  "Google Drive"
00000000`004bcf2a  "ProductVersion"
00000000`004bcf48  "2.34.5075.1619"
00000000`004bcf6e  "VarFileInfo"
00000000`004bcf8e  "Translation"
```

4.1.1.2 Data Storage Requirements

The application needs to capture date from SQLite and store it into MySQL. The data needs to be dated appropriately. The solution is required to run independently on each client. Central storage is required to capture data from different clients and store. Data is stored as system logs for easier retrieval. The logs are stored for retrieval and review if required.

4.1.1.3 Data Classification Requirements

The entire SQLite data can be captured and stored, but for logging purposes not all data is required. There needs to be some elimination of unwanted data and some form of "normalization" to remove unnecessary data from log files. The data available through digital forensics is in the following table representations; Table 4.2, Table 4.3, Table 4.4, Table 4.5 and Table 4.6 within the SQLite database:

Table 4.2 contains information about files uploaded into the cloud.

Table 4.2: Table Cloud_entry

Column	Data type
Doc_id	Text (primary key)
Filename	Text
Modified	Integer
Created	Integer
Acl_role	Integer
Doc_type	Integer
Removed	Integer
Size	Integer
Checksum	Text
Shared	Integer
Resource type	Text
Original size	Integer
Original checksum	Text

Column definitions:

Doc_ID: the first half of the characters for this field remains constant for every file uploaded by a user in their own Google Drive. The second half of characters keeps changing. The first 13 characters are similar for all files uploaded under the same user account even using different computers. An assumption can be made that the first half is attributed to the user account used to upload the document into Google drive. It can be used to uniquely identify a user account with a file. The Doc_ID also serves as the link via HTTP to the file. The Doc_ID utilizes a numeral system, which seems to be auto generated. Files created online seem to have longer Doc_IDs and do not seem to follow any pattern. Uploaded files have 28 character names (there were some few exceptions) while files created online have longer Doc_IDs more than 28 characters.

Filename: Actual filename of the file in the cloud sync folder.

Modified: Last date modified. This is in Unix timestamp; the number of seconds since 1 Jan 1970 at the time of modification.

Created: the date the file was created in the cloud, this field will remain empty if a file is created locally and uploaded into the cloud.

Acl_role: this column defines the creator of the document, files that have been created and shared by other users then downloaded into Google drive should have a value of 1. Files uploaded or created by a user in their own Google Drive display a value of 0.

Doc_type: This column should assign documents values based on the key below. However, it does not seem to work for documents created locally and uploaded. Mostly documents created and uploaded are just assigned 1 from the key below but document generated using Google Doc are appropriately assigned the other values below.

Document Type List:

i. 0 = place holder for folders
ii. 1 = Appears to be a place holder type for various file extensions. All files uploaded to the drive have this number.
iii. 2 = Google Presentation/ slides
iv. 3 = Google Form
v. 4 = Google Spreadsheet
vi. 5 = Google Drawing
vii. 6 = Google Document
viii. 12 = Google Map
ix. 13 = Google Site

Removed: all tests on this field returned negative results. The value remained 0, no assumptions or conclusion can be made.

Size: Size of the file. Folders do not appear to have values even if there are files inside them

Checksum: MD5 hash of the files. When files are created in the cloud they do not appear to get an MD5 hash. They get MD5 hash if they are locally placed in the Google Drive or uploaded via the web through the upload feature Google has.

Shared: shared files and folders are assigned 1, those not shared are 0 –a representation of Boolean true or false.

Resource_type: Files uploaded are only defined as files. Folders are appropriately named folder whether created offline or online. Files created online are defined as document.

Original_size: all test done returned negative results the field remained Null therefore no assumptions or conclusions can be made.

original_checksum: all test done returned negative results the field remained Null therefore no assumptions or conclusions can be made.

Table 4.3 contained information about files stored locally.

Table 4.3: Table Local_entry

Column	Data type
Inode	Integer (primary key)
Volume	Text
Filename	Text
Modified	Integer
Checksum	Text
Size	Integer
Is_folder	Integer

Column definitions:

Inode_number: Unique inode number assigned to each file. Under the local_relations table, it refers to the child_inode_number and connects to the parent_inode_number. The assumption is that it is a pointer reference to the file.

Volume: this column represents the volume serial in decimal. By running the command Vol C: in windows the same value in hexadecimal is retrieved. The value is the same as all files are stored locally on the same volume.

Filename: Actual filename of the file in the local default sync folder.

Modified: Last date modified. This is in Unix timestamp, i.e. the number of seconds since 1 Jan 1970 when file was modified.

Checksum - MD5 checksum of the file, as per calculated in the local default sync folder of the computer.

Size - File size measured in bytes.

Is_folder: This column defines whether a resources is a file or a folder. File is 0 and folder is 1.

Table 4.4 contained references between files and folders in the cloud.

Table 4.4: Table Cloud_relations

column	**Data type**
Child_doc_id	Text
Parent_doc_id	Text

Column Definitions:

Child_doc_id: references the doc_id of the file

Parent_doc_id: references the doc_id of the folder

Table 4.5 contained reference between files, folders and disk drive/ volume

Table 4.5: Table Local_relations

Column	**Data type**
Child_inode	Integer
Child_volume	Text
Parent_inode	Integer
Parent_volume	Text

Column Definitions:

Child_inode: references the file inode.

Child_volume: serial of local volume

Parent_inode: references the folder inode

Parent_volume: serial of local volume.

Table 4.6 contained reference found in other tables.

Table 4.6: Table Mappings

Column	**Data type**
Inode	Integer
Volume	Text
Doc_id	Text

These columns are already represented in other tables.

Two tables; Table volume_info and Table_overlay_status remain empty and did note populate any data.

4.1.1.4 Data Output

From the tables above the following information, displayed in Table 4.7, is derived to form logging information for Google Drive.

Table 4.7: Logging Output

Column	From table?
Doc_id	Cloud_entry
Filename	Local_entry
Checksum	Local_entry
Share	Cloud_entry
Acl_role	Cloud_entry
Modified	Cloud_entry
Volume	Local_entry

This requires that an operation to join the tables is conducted and appropriate primary keys and foreign keys identified in the tables.

Join operation involves cloud_entry + local_entry

4.1.1.5 Digital Artefacts Explained

Doc_id: This field can be used to map files to users. Using the first 13 characters it is possible to identify the person who originally uploaded a document to the cloud. This can only be done when looking for users within the Organization.

Filename: this can be used to identify the file and document type using the extension. Getting this data from the table local_entry can also help identify both windows created and Google Docs created documents. Files created using in Google Drive have the extension .gdoc, .gmap, .gform, et cetera. while local files only have normal windows extensions.

Checksum: this MD5 hash can be used to uniquely identify a resource. This data is collected from local_entry because in the cloud_entry table files created online do not have this field populated.

Share: this field will inform whether a resource is shared or not.

Acl_role: this field will indicate whether the file was created by the users or downloaded from another Google Drive shared resource.

Modified: this is the only populated time stamp. It provide information when the file was last accessed. This information is picked from cloud entry to so that online access can also be recorded.

Volume: this column shows under which volume the file currently resides on local device.

For the purposes of such an application some artefacts outside Google Drive are also important, the computer name and the timestamp of when any logging information is collected.

4.1.1.6 Application

The application runs silently on system shutdown or logoff and system restart, grabbing required information and appropriately storing in log files. The application does not require input from user but is set to run by an administrator and always run in the background silently. The application should collect information and store them appropriately into a MySQL database.

4.1.2 Functional Requirements

The system should have the following capabilities:

i. Capture and store required information into MySQL databases
ii. Automate process of replication to capture and create log files
iii. Capture only appropriate data into one log files each computer having its own database and filing daily logs into tables

Figure 4.1 presents complete overview of the proposed system.

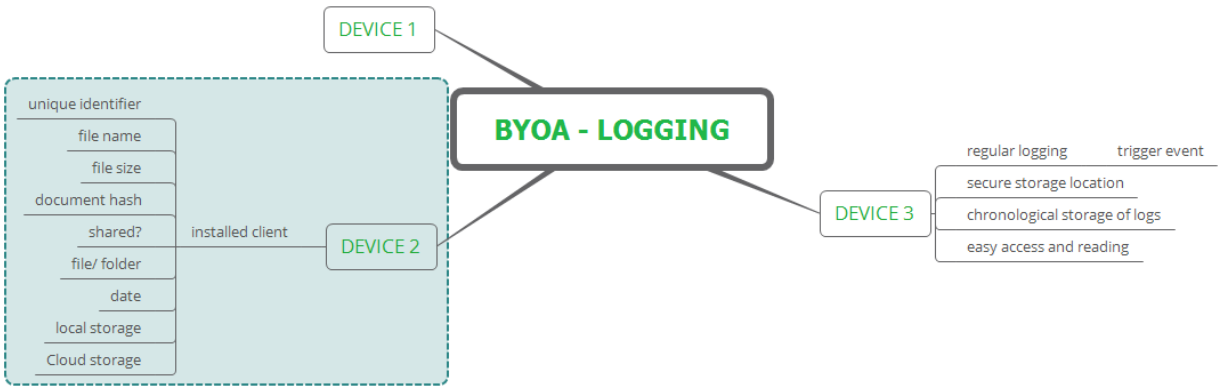


Figure 4.1: Proposed System Mind Map

4.2 System Design

Figure 4.2 shows all information available for capture. However, the information is more than required as per the system requirements.

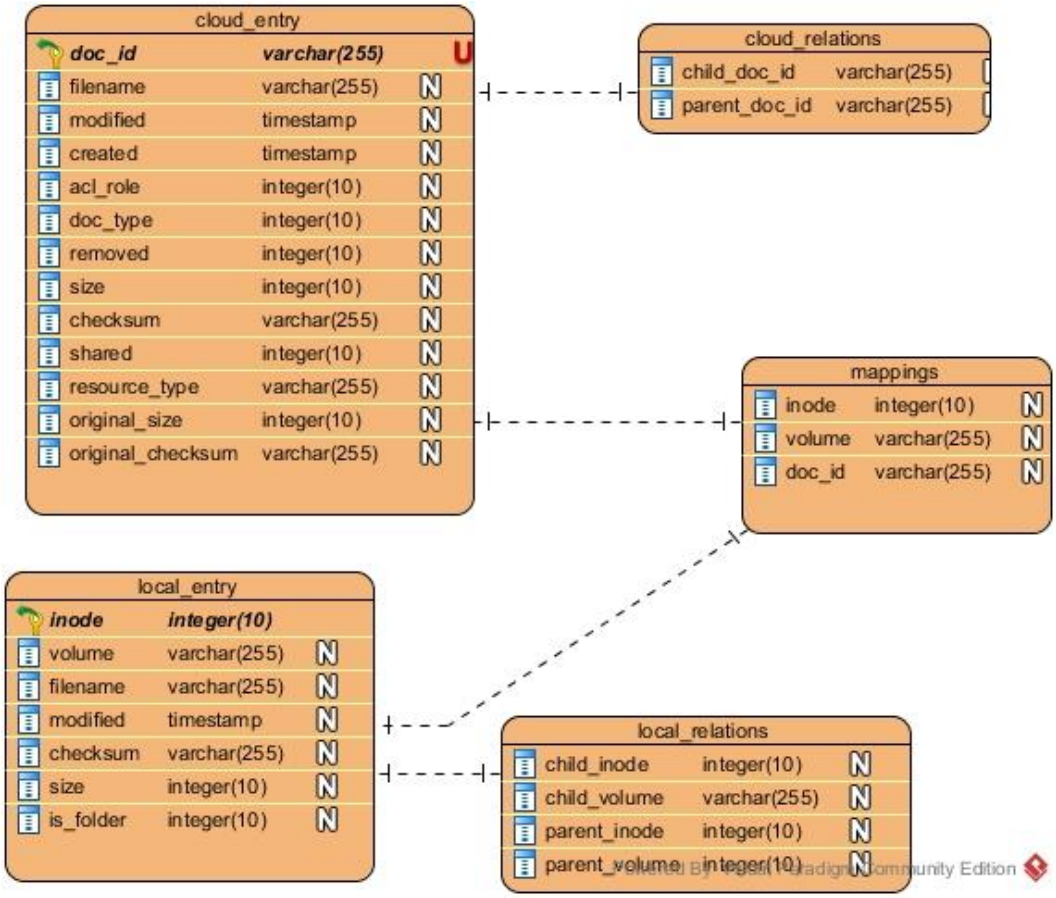


Figure 4.2: SQLite Tables

The following operations are conducted to capture the required data.

SELECT <select list>

FROM Table A A

FULL JOIN Table B B

ON A.key = B.key

Figure 4.3 displays this operation.

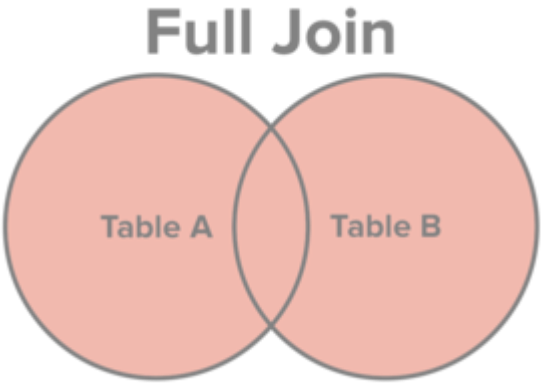


Figure 4.3: Join Tables

Figure 4.4 shows the information that is derived from the cloud_entry and local_entry to form the logging table.

Figure 4.4: Logging Table

The tables local_entry and cloud_entry are joined but the unrequired columns are left out to produce a log.

The best way to capture the logs is through a trigger event. The trigger event should activate a background application, which silently creates the necessary log files. This event must be constant to ensure that the logs are created. The best trigger event is system shutdown or logoff or restart. The sequence to create the logs is captured in the Figure 4.5.

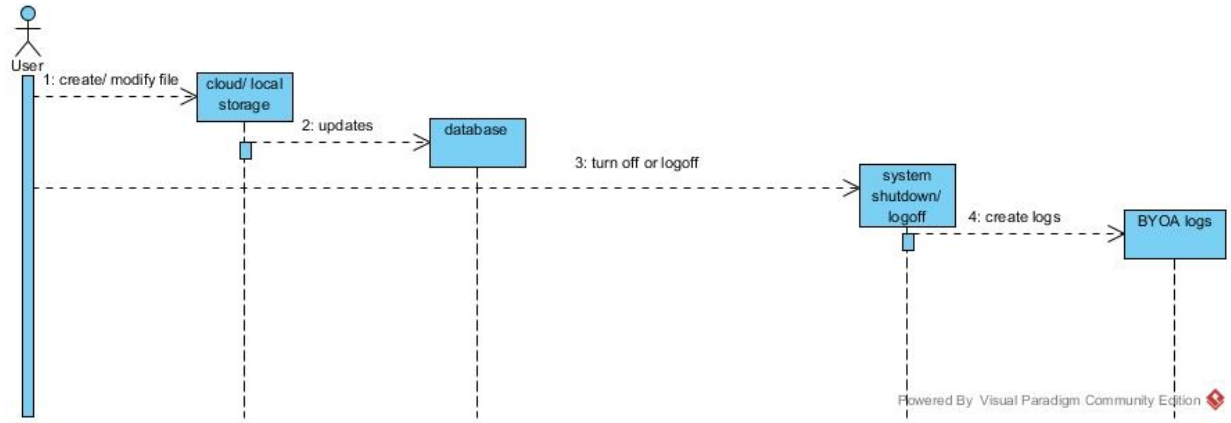


Figure 4.5: Sequence Diagram

The system should pick data from SQLite database, create appropriate logs in a CSV file and then transfer and store this data into MySQL. CSV is used as an intermediary to perform join actions from the required tables so that the data can be stored appropriately into MySQL.

Chapter 5: System Implementation and Testing

This chapter cover the process of building the application and testing it. It involves writing code to achieve the system design. The final deliverable is a working code/ application.

5.1 Implementation

The system is built using python programming language. This system/application does not have a graphical user interface, but rather it is expected to run silently in the background. The output it generates is stored for future review and analysis.

5.1.1 Python Implementation Description

Step 1: Data Acquisition

The first operation is the capture of required information, this is done by joining of two table; cloud_entry and local_entry and only selecting the required information fields. This is achieved by capturing the data through creation of a CSV file.

Comments to explain code are denoted by # and highlighted in grey.

Python Code: capture data from SQLite tables and store into CSV file

```
def main():
        #set the path where the Google information is stored
        googleDB = os.path.join("c:/users/" + os.environ.get("username") +
"/AppData/local/Google/Drive/user_default/snapshot.db")
        #connect to the SQLite file and extract data from the various tables and output into one
file
        try:
                conn = sqlite3.connect(googleDB, detect_types=sqlite3.PARSE_DECLTYPES)
                c = conn.cursor()
                c.execute("SELECT a.doc_id, a.shared, a.resource_type, a.acl_role, a.modified, b.checksum,
b.filename, b.volume FROM cloud_entry AS a JOIN local_entry AS b JOIN mapping AS c ON a.doc_id =
c.doc_id AND b.inode = c.inode")
        except Exception, e:
                        f = open('log.txt' , 'w')
```

```
                    #just in case there is a failure and error log is created in local computer
                    f.write("Google Drive not installed - confirm installation, and change file path for sqlite database in file googleEntries.py")
                    f.close()
        else:
            if not os.path.isdir('c:/byoa'):
                    os.mkdir('c:/byoa')
            with open('db.csv', 'wb') as f:
                    # creates the CSV file in current location
                    writer = csv.writer(f)
                    writer.writerow(['Document ID', 'Shared', 'Resource Type', 'Creator', 'Date Modified', 'Checksum', 'File Name', 'Volume Serial'])
                    writer.writerows(c)
if __name__== "__main__":
        main()
```

Step 2: Create Database in MySQL

It is important to store data in manner that it can be accessed and searched easily. If Google Drive in running in several computers, this information needs to be captured from all the computers. This application should have the capacity to create storage and appropriately store the logs for easy reference. Each computer should have its own database within MySQL server and store daily logs.

Python Code: Create MySQL database and name it using computer name (of the device)

```
import MySQLdb
import os
database = os.environ.get("computername")
#get the computer name to be used for appropriately naming the database
from warnings import filterwarnings
```

```
filterwarnings('ignore', category = MySQLdb.Warning)
```

```
# ignore MySQL warning, as code is meant to run silently, this is helpful when trying to create a database that already exists.
```

```
if __name__ == '__main__':
    connect = MySQLdb.connect(host="localhost", port=3306, user="root", passwd="")
```

```
# connect to MySQL server
```

```
cursor = connect.cursor()
cursor.execute("""CREATE DATABASE IF NOT EXISTS """ + str(database) + """;""" )
connect.commit()
connect.close()
```

Step 3: Create Table and Insert Data

For purposes of daily logging a table is created with an appropriate name (to make easy reference the current date is used) to store the data. The data is then transferred from the CSV file into the MySQL table.

Python Code: Create table in MySQL and insert data from CSV

```
import os
import re
import sys
import csv
import time
import argparse
import collections
import MySQLdb
import warnings
```

```
# ignore MySQL warnings
warnings.filterwarnings(action='ignore', category=MySQLdb.Warning)
# use time stamp to name table – for easier reference
log = time.strftime("%Y%m%d")
str_log = ("m"+ str(log)+"m")

#get data type from the various columns in CSV file
def get_type(s):
    """Find type for this string
    """
    # try integer type
    try:
        v = int(s)
    except ValueError:
        pass
    else:
        if abs(v) > 2147483647:
            return 'bigint'
        else:
            return 'int'
    # try float type
    try:
        float(s)
```

```
except ValueError:
    pass
else:
    return 'double'


 # check for timestamp
dt_formats = (
    ('%Y-%m-%d %H:%M:%S', 'datetime'),
    ('%Y-%m-%d %H:%M:%S.%f', 'datetime'),
    ('%Y-%m-%d', 'date'),
    ('%H:%M:%S', 'time'),
)
for dt_format, dt_type in dt_formats:
    try:
        time.strptime(s, dt_format)
    except ValueError:
        pass
    else:
        return dt_type


# does not match any other types so assume text
if len(s) > 255:
    return 'text'
else:
```

```
        return 'varchar(255)'



def most_common(l, default='varchar(255)'):
    """Return most common value from list
    """
    # some formats trump others
    if l:
        for dt_type in ('text', 'bigint'):
            if dt_type in l:
                return dt_type
        return max(l, key=l.count)
    return default



def get_col_types(input_file, max_rows=1000):
    """Find the type for each CSV column
    """
    csv_types = collections.defaultdict(list)
    reader = csv.reader(open('db.csv'))
    # test the first few rows for their data types
    for row_i, row in enumerate(reader):
        if row_i == 0:
            header = row
```

```
        else:
            for col_i, s in enumerate(row):
                data_type = get_type(s)
                csv_types[header[col_i]].append(data_type)

        if row_i == max_rows:
            break

    # take the most common data type for each row
    return [most_common(csv_types[col]) for col in header]


def get_schema(str_log, header, col_types):
        schema_sql = ("""CREATE TABLE IF NOT EXISTS """ + str_log + """ (document_ID VARCHAR(255), shared INT(11), resource_type TEXT, creator INT(11), date_modified DATE, checksum TEXT, filename TEXT, volume INT(11));""")
        return schema_sql


def get_insert(str_log, header):
    """Generate the SQL for inserting rows
    """
    field_names = ', '.join(header)
    field_markers = ', '.join('%s' for col in header)
    return 'INSERT INTO %s (%s) VALUES (%s);' % \
```

```
    (str_log, field_names, field_markers)



def format_header(row):
    """Format column names to remove illegal characters and duplicates
    """
    safe_col = lambda s: re.sub('\W+', '_', s.lower()).strip('_')
    header = []
    counts = collections.defaultdict(int)
    for col in row:
        col = safe_col(col)
        counts[col] += 1
        if counts[col] > 1:
            col = '{}{}'.format(col, counts[col])
        header.append(col)
    return header



def main(max_inserts=10000):
        print "Importing `%s' into MySQL database `%s.%s'" % ('db.csv', database, str_log)
        db = MySQLdb.connect(host="localhost", port=3306, user="root", passwd="", db=database,
charset='utf8')
        cursor = db.cursor()
        # create database and if does not exist
```

```
cursor.execute('CREATE DATABASE IF NOT EXISTS %s;' % database)

db.select_db(database)


# define table

print 'Analyzing column types ...'

col_types = get_col_types('db.csv')

print col_types


header = None


for i, row in enumerate(csv.reader(open('db.csv'))):

        if header:

                while len(row) < len(header):

                        row.append('') # this row is missing columns so pad blank values

                cursor.execute(insert_sql, row)

                if i % max_inserts == 0:

                        db.commit()

                        print 'commit'

        else:

                header = format_header(row)

                schema_sql = get_schema(str_log, header, col_types)

                print schema_sql

                # create table

                cursor.execute('DROP TABLE IF EXISTS %s;' % str_log)
```

```
        cursor.execute(schema_sql)
        # create index for more efficient access
        try:
            cursor.execute('CREATE INDEX ids ON %s (id);' % str_log)
        except MySQLdb.OperationalError:
            pass # index already exists

        print 'Inserting rows ...'
        # SQL string for inserting data
        insert_sql = get_insert(str_log, header)

    # commit rows to database
        print 'Committing rows to database ...'
        db.commit()
        print 'Done!'
if __name__ == '__main__':
        main()
```

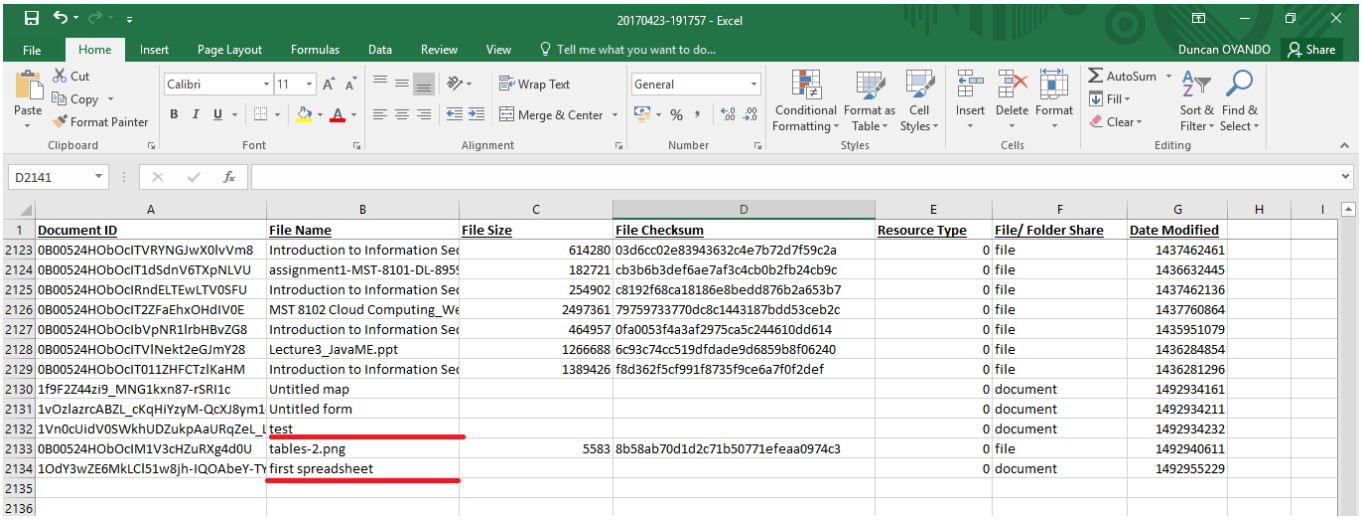
5.2 System Testing

The testing approach takes two forms; one is the testing of the application and whether it can efficiently run and capture the required data and the other is feedback from potential users on whether the tool adds value and improves the security setup in a BYOA environment.

5.2.1 Functional Testing

To ensure the system function ran as required various scenarios are simulated as below and results recorded:

Scenario one: Creation of the CSV file, this test is done to ensure the code for creating the file works as required. Also there is need to ensure that file activity is captured on the CSV file. One file is copied into the local Google Drive repository and another is created online. The file copied is a JPEG: ‘’tables-2.jpg and the file online created is ‘first spreadsheet’. Once the files have synchronised/ replicated the python script for creating a CSV is executed. Figure 5.1 shows the results of the test.



	A	B	C	D	E	F	G	H	I
1	Document ID	File Name	File Size	File Checksum	Resource Type	File/ Folder Share	Date Modified		
2123	0B00524HObOcITVRYNGJwX0lvVm8	Introduction to Information Sec	614280	03d6cc02e83943632c4e7b72d7f59c2a	0	file	1437462461		
2124	0B00524HObOcIT1dSdnV6TXpNLVU	assignment1-MST-8101-DL-895	182721	cb3b6b3def6ae7af3c4cb0b2fb24cb9c	0	file	1436632445		
2125	0B00524HObOcIRndELTEwLTV0SFU	Introduction to Information Sec	254902	c8192f68ca18186e8bedd876b2a653b7	0	file	1437462136		
2126	0B00524HObOcIT2ZFaEhxOHdIV0E	MST 8102 Cloud Computing_We	2497361	79759733770dc8c1443187bdd53ceb2c	0	file	1437760864		
2127	0B00524HObOcIbVpNR1IrbHBvZG8	Introduction to Information Sec	464957	0fa0053f4a3af2975ca5c244610dd614	0	file	1435951079		
2128	0B00524HObOcITVlNekt2eGJmY28	Lecture3_JavaME.ppt	1266688	6c93c74cc519dfdade9d6859b8f06240	0	file	1436284854		
2129	0B00524HObOcIT011ZHFCTzlKaHM	Introduction to Information Sec	1389426	f8d362f5cf991f8735f9ce6a7f0f2def	0	file	1436281296		
2130	1f9F2Z44zi9_MNG1kxn87-rSRI1c	Untitled map			0	document	1492934161		
2131	1vOzlazrcABZL_cKqHiYzyM-QcXJ8ym1	Untitled form			0	document	1492934211		
2132	1Vn0cUidV0SWkhUDZukpAaURqZeL_L	test			0	document	1492934232		
2133	0B00524HObOcIM1V3cHZuRXg4d0U	tables-2.png	5583	8b58ab70d1d2c71b50771efeaa0974c3	0	file	1492940611		
2134	1OdY3wZE6MkLCl51w8jh-IQOAbeY-TY	first spreadsheet			0	document	1492955229		
2135									
2136									

Figure 5.1: Test Results 1

The CSV file is created and the files are found in the CSV file details. The red lines display demarcate where the file names of the files created.

Scenario 2: Involves testing the code that creates the database and tables in MySQL server. The code for creating a database and the code for creating a table are executed sequentially. Figure 5.2 shows the test results.

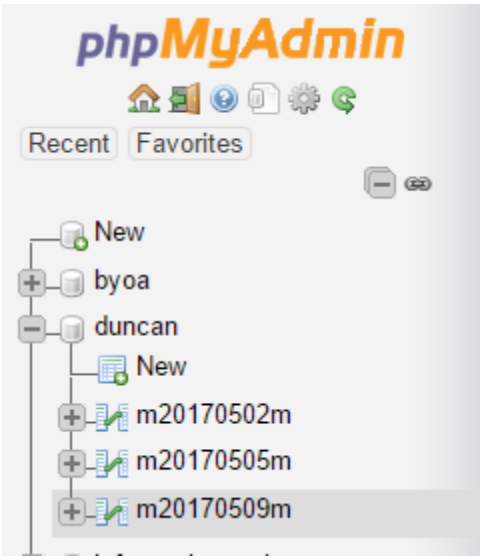
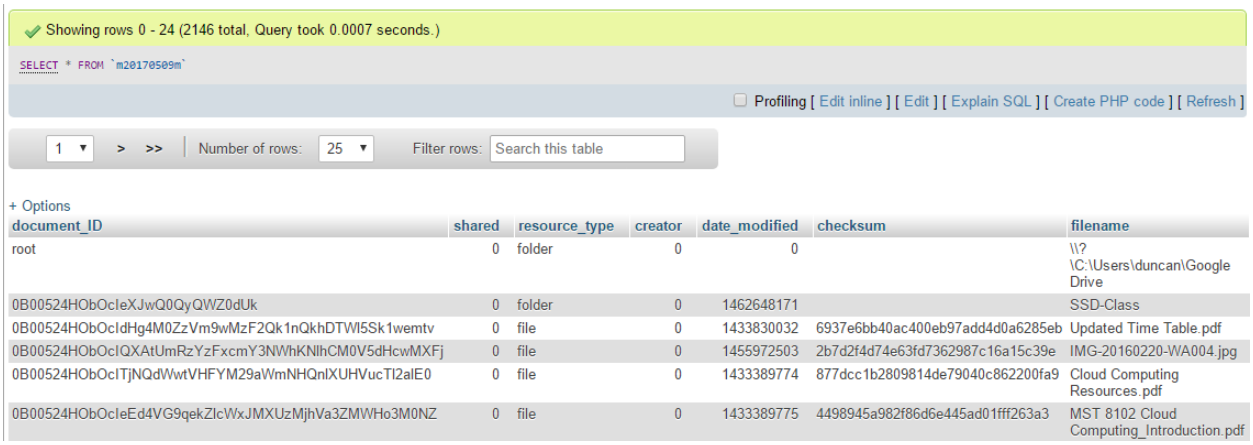


Figure 5.2: Test Results 2

A look into the MySQL server show that a database similar to the computer name has been created and three tables within the database named using the format *m*YYYYMMDD*m.* A look into the tables also show that data is appropriately captured. Figure 5.3 shows the table in MySQL.



document_ID	shared	resource_type	creator	date_modified	checksum	filename
root	0	folder	0	0		\\?\C:\Users\duncan\Google Drive
0B00524HObOcIeXJwQ0QyQWZ0dUk	0	folder	0	1462648171		SSD-Class
0B00524HObOcIdHg4M0ZzVm9wMzF2Qk1nQkhDTWl5Sk1wemtv	0	file	0	1433830032	6937e6bb40ac400eb97add4d0a6285eb	Updated Time Table.pdf
0B00524HObOcIQXAtUmRzYzFxcmY3NWhKNlhCM0V5dHcwMXFj	0	file	0	1455972503	2b7d2f4d74e63fd7362987c16a15c39e	IMG-20160220-WA004.jpg
0B00524HObOcITjNQdWwtVHFYM29aWmNHQnlXUHVucTI2alE0	0	file	0	1433389774	877dcc1b2809814de79040c862200fa9	Cloud Computing Resources.pdf
0B00524HObOcIeEd4VG9qekZlcWxJMXUzMjhVa3ZMWHo3M0NZ	0	file	0	1433389775	4498945a982f86d6e445ad01fff263a3	MST 8102 Cloud Computing_Introduction.pdf

Figure 5.3: MySQL Table

A total of 2146 entries are captured and also the data show the appropriate information for every field as represented in the SQLite database of Google Drive.

The snapshot of a table data dictionary also validates the data types in the python code for creating the table this is shown in Figure 5.4.

m20170509m

Column	Type	Null	Default
document_ID	varchar(255)	Yes	*NULL*
shared	int(11)	Yes	*NULL*
resource_type	text	Yes	*NULL*
creator	int(11)	Yes	*NULL*
date_modified	int(11)	Yes	*NULL*
checksum	text	Yes	*NULL*
filename	text	Yes	*NULL*
volume	int(11)	Yes	*NULL*

Figure 5.4: Data Dictionary

Scenario 3: This test case scenario check the status of the database before and after execution of the application. Currently in the MySQL database there is a database 'duncan' appropriately named from the computer name with one table captured 1st Nov 2017 (20171101) as per the Figure 5.5.

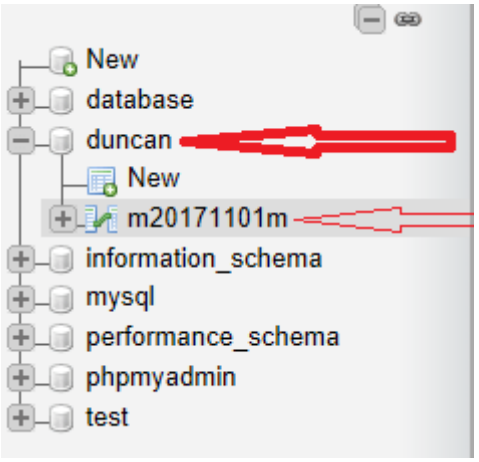


Figure 5.5: Database Before

A scenario of the application was executed on 2nd Nov 2017 and the results are captured as per Figure 5.6 and Figure 5.7.

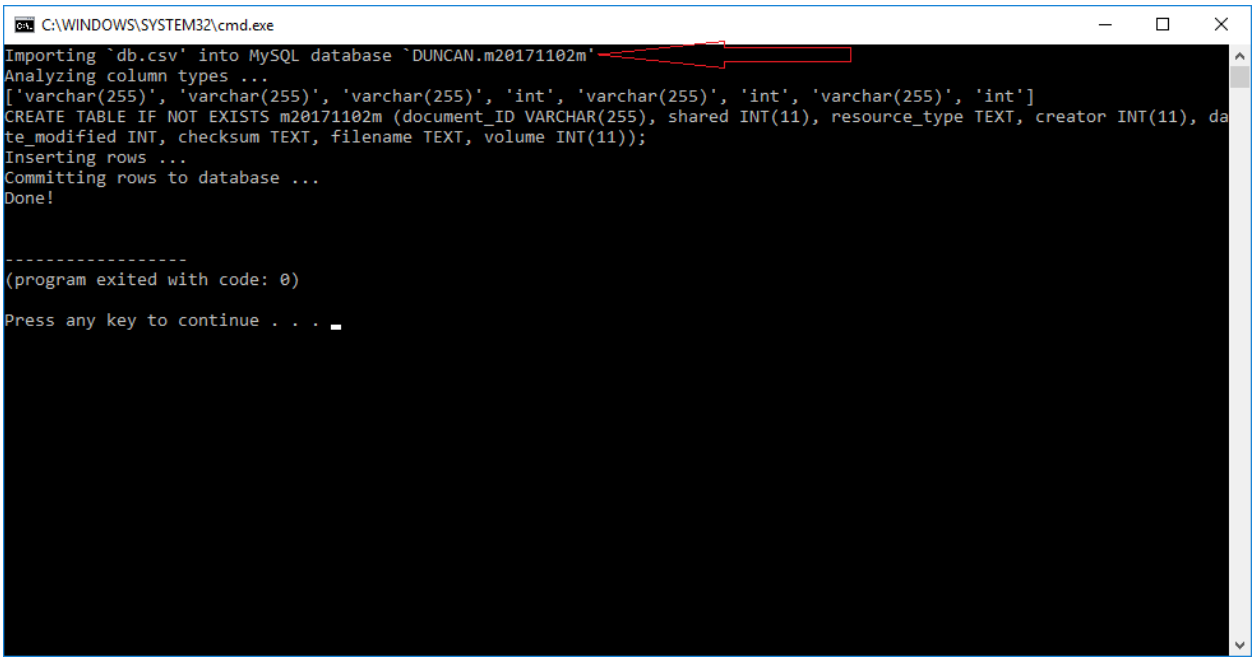


Figure 5.6: Code Execution

Note database name and table name as indicated by red arrow. Also, a comparison can be made from the columns names and types in figure 5.4 and 5.6. They are exactly the same.

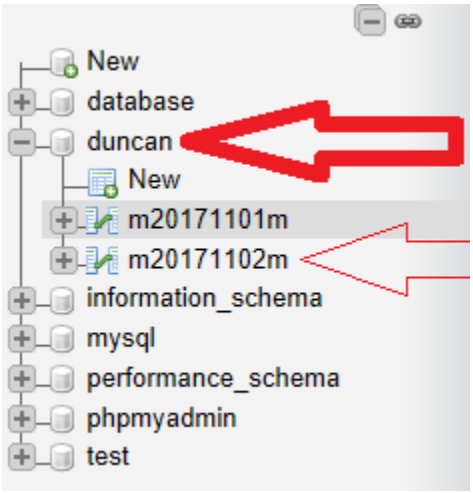


Figure 5.7 : Database After

Note the additional table created and appropriately named.

Scenario 4 : This test was conducted by running the application in a different computer. The expected results were the creation of a new Database in the MySQL server and new table according to the date of execution which is 2nd November 2017. Figure 5.8 shows the

old computer database with a red arrow and new computer database with a green arrow and the new table appropriately named.

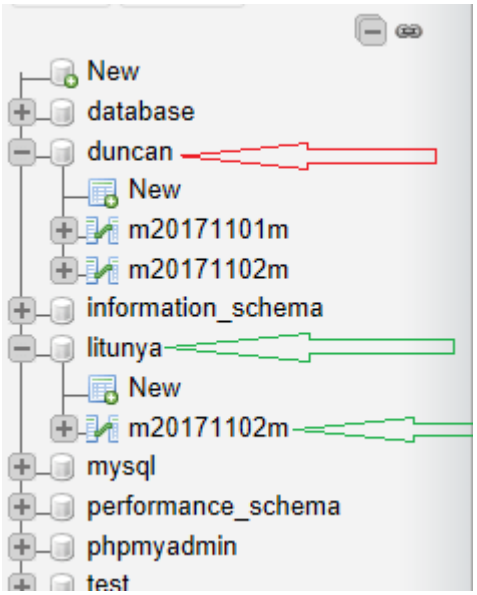


Figure 5.8 : New Database

5.2.2 System Validation

To ensure that the system meets the demands or adds value in securing the situation in BYOA environment, the application is shared with two IT managers for their feedback in the following areas;

i. Does the solution offer enhanced security on BYOA: Google Drive?
ii. Do the functions within the solution cover all aspects and if not what areas are missing?

Feedback: the applications enhances the logging capabilities. Even if the client application is uninstalled, there is the possibility to gather information from another source. The database captures information in a repetitive manner in tables.

Queries can be done to find out what files have been deleted from the various Google Drive accounts even if something happens to the SQLite database or the Google Drive Client is uninstalled. Interference with the Google Docs activity logs can also be identified as there are multiple logs created.

Centralised storage of all logs via MySQL server is also good, makes it easier for an analysis to be conducted. The analysis of Google Drive artefacts provides guidance while evaluating the log file. It is easy to know files created online and files created locally, it is possible to trace a file upload to a specific account – but limited to accounts used within the organisation.

It possible to identify which resources are shared and those that are not and also possible to know if a file was created/ uploaded by a user or just downloaded from another shared Google Drive. The solution provides adequate information for logging file activity in Google Drive.

Chapter 6: Discussion of Results

6.1 Introduction

The goal of this research was to understand the security and digital forensics challenges in a BYOA cloud applications, analyse available tools and techniques that enhance security then develop and validate a digital forensics application for Google Drive that can be used to enhance security.

This was done with the aim of providing knowledge of what digital forensics artefacts are available from the use of Google Drive and also provide insight into how such artefacts can be used to improve security. The targeted users are SMEs which do not have enough financial muscle for other security options available but would like to leverage on the free to public BYOA that can boost the operations.

6.2 Explanation of Findings

The focus group sessions were used to define system requirements. Based on their experiences the participants were able to determine what would be ideal for capturing and logging for security purposes. Further research on Google Drive digital forensics artefacts was done to determine what would make the correct fit for these requirements in order to get the right information logged by the application. The focus group sessions were complimented by digital forensics research into Google Drive to ensure the correct information is captured by the system.

6.3 Discussion

The first objective was to investigate factors relating to security in BYOA cloud applications. This objective was achieved through Literature Review in Chapter 2, a thorough study was done to understand BYOA, their adoptions in the modern office and security issues around their adoption and usage. Digital Forensics as a technique for security is also discussed. Google Drive as a BYOA was also investigated in depth looking into challenges in terms of security.

The second objective; analyse available tools and techniques that provide security for BYOA cloud applications, was also covered in the Literature Review in Chapter 2. In Section 2.3, the issue of governance of BYOA is explored. The sections takes a look at considerations that have to be taken into account while implementing BYOA. It takes a look at how to control the BYOA implementation. Section 2.3 also looks at the best areas where security for BYOA can be

implemented. In Section 2.4, available tools and methods for security are discussed and evaluated, though there are options available a clear gap for SMEs and organisations with limited budgets is identified. Section 2.5 discusses the techniques for digital forensics and the role it plays in IT security. The role of data acquisition and data organisation for purposes of security are discussed.

The third objective was to develop a digital forensics applications to enhance security in BYOA cloud applications. The foundation for this objective is laid down in chapter 2. Section 2.5 clearly brings out the role of incident response and retrospective analysis. This objective is further achieved in Chapter 3, 4 and 5. In Chapter 3, an appropriate system development methodology and research technique were used to gather system requirements for a solution that can address security concerns when implementing BYOA (Google Drive). In Chapter 4, digital forensics artefacts were identified, how they can be obtained specifically for Google Drive and how they can be used. A study of the digital forensics artefacts is done to identify how the artefacts can be useful in securing the BYOA environment.

These artefacts are the core information for a logging application which can be used to enhance security. The application is developed as documented in Chapter 5 Section 5.1. The system undergoes some simple tests to ensure that it is functional as documented in Section 5.2.

The final objective is to validate the developed solution. This objective is achieved in Chapter 5 Section 5.2.2. The proposed system after undergoing repetitive enhancements and tests is finally evaluated through feedback from potential users. The feedback is positive, the system can be used to enhance security through monitoring and logging of BYOA activities on devices.

6.4 Opportunities for this Approach

This approach to security is more relevant to SMEs. Organisation that leverage of free to public BYOA don't have enough financial capacity to implement the security solutions on offer. These organisations also have IT and IT security rolled into one or probably under the same person. In such situations innovative approaches are required to find solutions that don't require huge financial investments. This solution can be ran from a simple server only requiring MySQL server which is freely available and can operate in LAN or over the internet. The clients only need Python interpreter installed. With this simple set up, clients can begin sending their data to the server where the data is stored under the device name and according to dates.

6.5 Limitations of this Application

The solution developed is for a Windows platform. It is meant to improve security but not offer total security. It places emphasis on obtaining and storing information that can be used for security purposes; intelligence gathering and investigations. This means that in most cases it may offer insight after an incident has occurred.

Chapter 7: Conclusion, Recommendations and Future Work

7.1 Conclusion

This research reviews the security challenges in BYOA, it looked into possible solutions that can be implemented to improve the security. BYOA is important to SMEs and organisation with limited budgets, this users offer the niche market for this type of solutions. This solution although built for Google Drive demonstrates a concept that can work with most free to public applications including browsers. It provides a simple solution to logging of activities of Google Drive, allowing multiple devices to be monitored. The information is easily retrievable from the database.

7.2 Recommendations

Logging for BYOA needs to incorporate BYOD: bring your own device. Methods for logging need to be able to run independent of the operating platform; phone or computer and store the logging information centrally via internet.

7.3 Future Work

This research also provides a deeper insight into digital forensics of Google Drive. By experimenting and analysing the artefacts some assumptions have been made that can aid in future forensics analysis. It also opens the door for future research into this area, to properly analyse these artefacts and come to a conclusion with solid evidence as to what they represent. Examples are; to find out how the document ID and inode numbers are generated and find how to link the document ID to a Google account. Such research can aid in digital forensics work of Google Drive.

There is still need to accurately investigate the purposes of the various columns and tables that store data in SQLite on Google Drive usage. While this research was able to identify some of the artefacts and the relevance of some of the columns, it is still not clear the relevance of the following and what information is stored in them or what that information means;

i. Table volume_info and Table overlay_status
ii. Columns in cloud_entry
    a. Removed
    b. Original size
    c. Original checksum

Although based on the names of the columns it is easy to make assumptions, specific tests done to trigger positive results based on the assumptions were done without producing results.

Also it is important to understand the generation of the following fields;

i. Table cloud_entry: Doc_id
ii. Table local_entry: Inode

There seems to be a relationship between the Doc_ID and the user account. If such a connection can be confirmed and the how the field is generated it may be possible to identify the user account using the Doc_ID.

Future research needs to take into consideration that Cloud Sync applications are now incorporating encryption. Dropbox already uses encryption – how do you log activity of such application? The same may apply on Google Drive soon and other BYOA cloud applications.

References

Akpose, W. (2014). *Cloud in the horizon: Opportunities and challenges for enterprises in the cloud economy.* 6igma Associates.

Bennett, J. (2016, February 26). *Strategies for the Bring Your Own App Surge.* Retrieved from Gartner: http://www.gartner.com/smarterwithgartner/bring-your-own-app-strategies/

Comcast Business View. (2016). *BYOD/BYOA: A Growing, Applicable Trend*. Retrieved from INC: http://www.inc.com/comcast/byod-byoa-a-growing-applicable-trend.html

Freiling, S. (2007). A Common Process Model for Incident Response and Digital Forensics. *IMF 2007 3rd International Conference on IT-Incident Management & IT-Forensics.* Sturttgart.

Garrison, C. (2010). *Digital Forensics for Network, Internet and Cloud Computing.* Elsevier.

Gartner. (2017). *Cloud Access Security Brokers (CASBs)*. Retrieved from Gartner: http://www.gartner.com/it-glossary/cloud-access-security-brokers-casbs/

Gottesdiener, E. (1995). *RAD Realities: Beyond the Hype to How RAD Really Works.*

Grance, P. M. (2009). *The NIST Definitation of Cloud Computing.* National Institute of Standards and Technology Laboratory.

Green, J. (2015). *Cyber Security: An introduction for Non-Technical Managers.* London: Routledge.

Log Me In. (2013). *Bring Your Own Application: The New Reality for Mobile Workforce.*

Martin, J. (1990). *Rapid Application Developement.*

Microsoft. (2014). *Manage Facebook contact sync in your organization*. Retrieved from Microsoft: https://blog.mypermissions.com/2014/07/15/your-enterprise-information-leaks-into-facebook-and-other-social-networks/

Mordhorst, M. (2014). *How to Help Emterprises Going Mobile.* Anchor Academic.

Moss, S. (2015, August 26). *4 Tips for Successful BYOA Governanc*. Retrieved from Enterprise Apps Today: http://www.enterpriseappstoday.com/management-software/4-tips-for-successful-byoa-governance.html

Neelankavil, J. P. (2007). *International Business Research.* New York: M.E. Shape, INC.

NIST. (2012). *Computer Security Incident Handling Guide.*

Osborn, C. (1995). *SDLC, JAD and RAD: Finding the Right Hammer*. Centre for Information Management Studies.

Parizo, C. (2016). *Prevent BYOA headaches with clear communication and careful monitoring*. Retrieved from Tech Target: http://searchcloudapplications.techtarget.com/tip/Prevent-BYOA-headaches-with-clear-communication-and-careful-monitoring

Patel, R. (2014). *Enterprise Mobility Strategy & Solutions.* Partridge.

Qing Li, G. C. (2015). *Security Intelligence: A Practitioner's Guide to Solving Enterprise Security Challenges.* Wiley.

Rouse, M. (2016). *bring your own apps (BYOA)*. Retrieved from TechTarget: http://searchsecurity.techtarget.com/definition/bring-your-own-apps-BYOA

Seth Early, R. H. (2014). From BYOD to BYOA, Phishing and Botnets. *IEEE Computer Society*, 16-18.

Shelly, G. (2009). *Systems Analysis and Design*. Cengage Learning.

Stuart, A. (2016). *BYOA: Challenges and caveats for controlling the flood of personal apps*. Retrieved from Tech Target: http://searchcloudapplications.techtarget.com/tip/BYOA-Challenges-and-caveats-for-controlling-the-flood-of-personal-apps

TechTarget. (2017). *Google Docs*. Retrieved from TechTarget: http://whatis.techtarget.com/

Walters, R. (2013). Bringing IT out of the Shadows. *Network Security*, 1-20.

Appendix A: Focus Group Notes

Introduction: The purposes of this discussion is to steer the development of a software solution that can capture and log information from the uses of Google Drive/ Google Docs with the objective of improving security in BYOA environment. The groups will guide the exploration and development based on the following:

- General security problems in an environment where Google Drive is in use
- What is considered useful information for capturing and logging.
- Proposed storage of the information captured.

Participants: IT managers who work for small companies ranging from 10 – 20 users that have Google Drive operating within their shadow IT can be selected as part of the focus group. Their knowledge of challenges on issues of IT security is considered valuable in designing a solution to address the problem.

Facilitator/ Moderator: The facilitator of the focus group will be the software developer. In order to steer the discussions in the right direction and ensure the objective is met.

Discussion Guide: Every session will have a main topic to ensure progress and structured discussions, though not as rigid as a questionnaire.

Meeting Records: notes will be taken and structured to reflect system development points. Where necessary mind maps will be produced to capture the essence of the discussions.

Time and place of sessions: Focus group sessions will be conducted via Skype. The timings of the meetings will be agreed by focus group participants availability with the guidance of the moderator.

Focus Group Session one

Discussion Topic: Information requirement for Security Logging: Google Drive

Welcome and thank you for volunteering to participate in my dissertation project, specifically to assist me by providing your input into the development of a software solution to aid in security where BYOA: Google Drive is in use. You have been selected as you all work in a similar setting with Google Drive operating in Shadow IT or operating without endorsement by IT and not blocked by IT.

Anonymity: Although no information will be collected from your respective office nor any software will be run in your office systems, the participants details will not be recorded. The assumptions is that you should share your personal opinion based on your personal experience and not the official IT policy of your respective office. The participants are free to know each other therefore is to follow so us to understand each other better. Introductions will not be captured on the official meeting records.

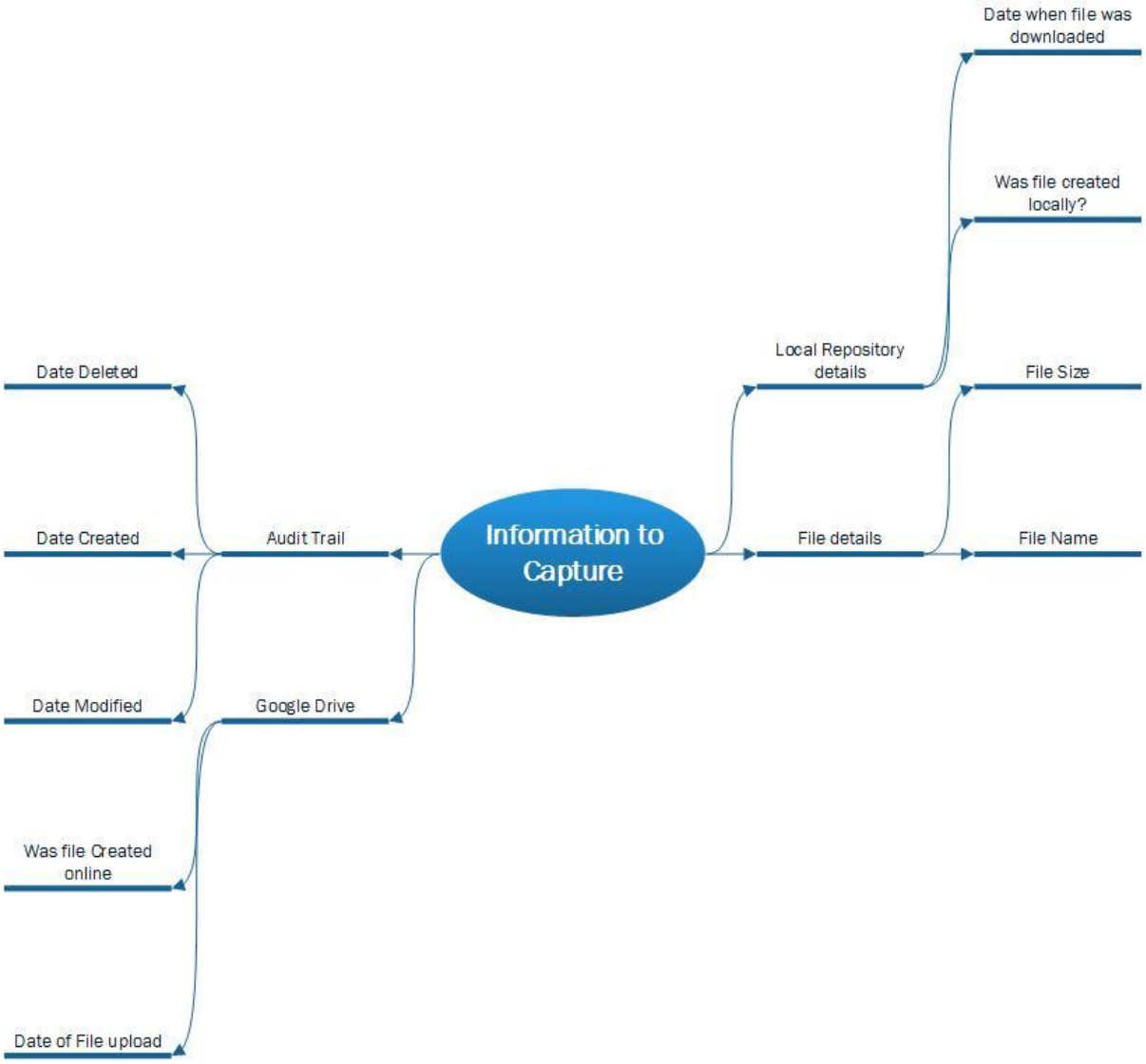
Ground Rules

- One person speaks at a time. Avoid interrupting another person.
- There are no right or wrong answers.
- There is no particular order of speaking, inform the moderator through chat if you want to speak next.
- I encourage everyone to share something, it is important that I obtain the views of each of you.
- You do not have to agree with the views of other people in the group.

Background: To bring everyone to speed, I will share a short summary of the project.

Today's discussion: based on the information shared, what are the most important details that should be captured on the usage of Google Drive?

Conclusion



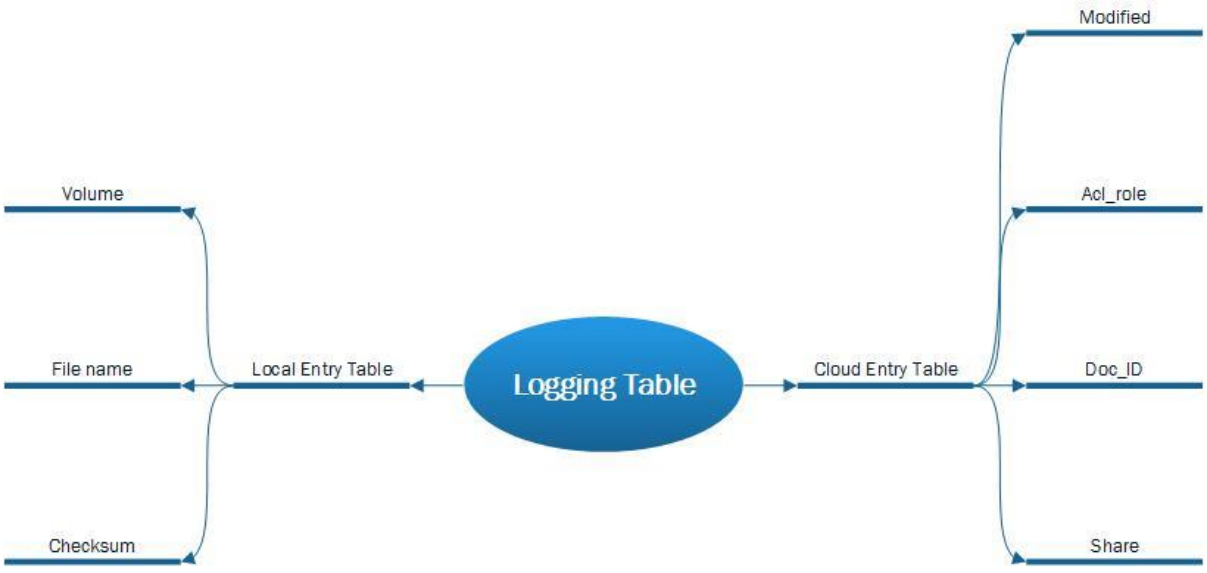
Focus Group Session Two

Discussion Topic: Comparing what is available from digital artefacts and what was identified from last discussions.

Welcome to the second meeting. Today we cover the requirements we identified from the last session with the information I have collected from my investigation on digital forensics artefacts available from Google Drive usage.

Conclusion

Column	Details	From table?
Doc_id	Unique identifier for file	Cloud_entry
Filename	File name as saved on repository	Local_entry
Checksum	MD5 hash of the files.	Local_entry
Share	Boolean value to indicate if file is shared	Cloud_entry
Acl_role	Boolean Value to indicate file owner	Cloud_entry
Modified	Unix time stamp for last date of modification	Cloud_entry
Volume	Volume Serial in Decimal	Local_entry

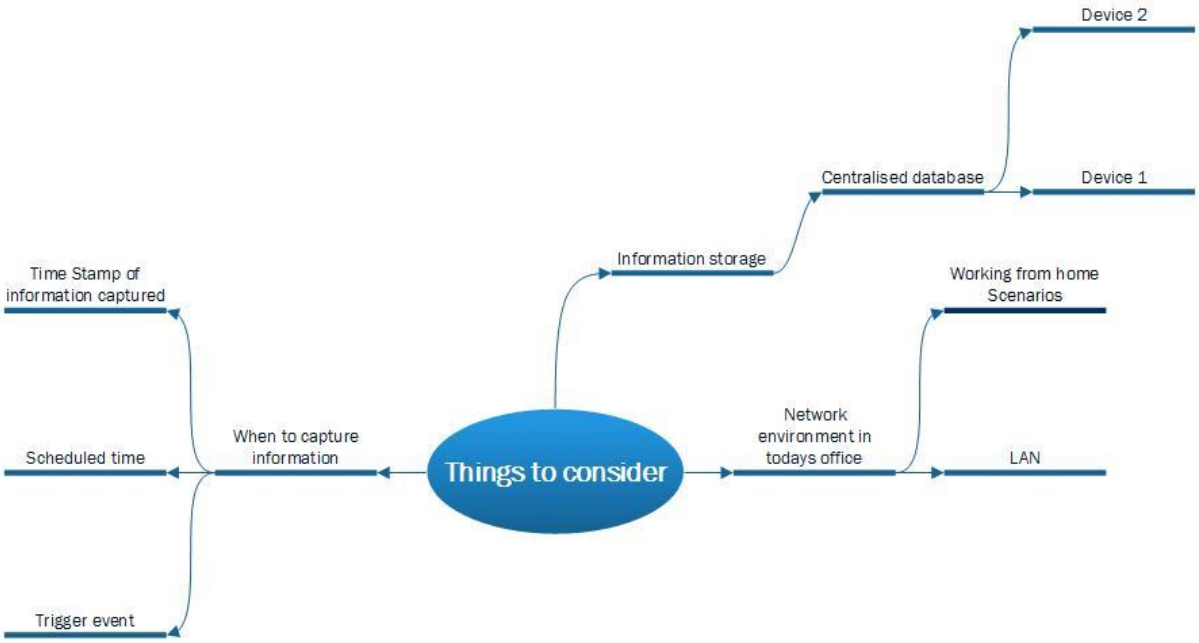


Focus Group Session Three

Discussion Topic: issues to consider when developing the application.

Welcome to the session, today we cover what other considerations should be made while developing the software solution.

Conclusions:



Appendix B: Validation Questionnaire

Dear Sir/ Madam,

I am currently undertaking a Master of Science degree in Information Systems Security at Strathmore University Nairobi, Kenya. In partial fulfilment of my course I am undertaking a dissertation research project into Security in Bring Your Own Application. The topic I have chosen is Securing a Bring Your Own Application Cloud Environment Using Digital Forensics. I have developed a software application that can collect logs from the devices using Google Drive. I would like to present this solution to you and explain how it works.

I would be very grateful if you could then, complete the questionnaire below. All information provided will be treated with strict confidence and individual firms will not be identified.

The questionnaire can be returned via email. I would be very grateful if you could complete within one working week.

Best regards

Duncan Oyando

Kindly answer the following questions;

1. Did you understand all the aspects of the solution and how it works?
2. Does the solution offer enhanced security on BYOA: Google Drive?
3. Do the functions within the solution cover all aspects of monitoring and logging and if not what areas are missing?

Questionnaire Feedback 1

1. Did you understand all the aspects of the solution and how it works?
   Yes.
2. Does the solution offer enhanced security on BYOA: Google Drive?

The solution enhances security by providing centralised storage services of Google Drive activities. Normally this activity is only available from user devices, and still at the mercy of the user depending on device settings. Valuable information can now be collected and stored in one location and also in manner that can be easily retrieved and also identified to a particular device.

3. Do the functions within the solution cover all aspects of monitoring and logging and if not what areas are missing?

   More information should be collected, also the other devices should be part of this like android phones.

Questionnaire Feedback 2

1. Did you understand all the aspects of the solution and how it works?

   Yes.

2. Does the solution offer enhanced security on BYOA: Google Drive?

   Storing all information in one database provides the capabilities of querying data to discover more where necessary. The application stores data in a structure manner making it easy to search and retrieve information. This application in an important cog in the wheel of IT security intelligence – it provides real time collection and normalization of generated by Google Drive storing it into a database ready to be analysed.

3. Do the functions within the solution cover all aspects of monitoring and logging and if not what areas are missing?

   When users are using applications like Google Drive, they will also use multiple devices. This application should consider this. It should be able to run in multiple applications, and on multiple Operating Systems.

Appendix C: Turnitin Originality Report

Processed on: 11-Apr-2018 10:31 AM EAT

ID: 944825395

Word Count: 13437

Submitted: 1

MISS 2018 By Duncan Litunya
Similarity Index: 15%
Similarity by Source:
Internet Sources: 13%
Publications: 2%
Student Papers: 9%
3% match (Internet from 13-Mar-2016)
http://searchwilderness.com
2% match (student papers from 11-Jul-2013)
Submitted to Middlesex University on 2013-07-11
1% match (Internet from 23-May-2014)
http://bitforensics.blogspot.co.uk
1% match (student papers from 19-Mar-2015)
Submitted to Strathmore University on 2015-03-19
1% match (Internet from 08-Jul-2012)
http://www.sysforensics.org
<1% match (Internet from 25-Apr-2016)
http://blog.mypermissions.com
<1% match (Internet from 09-Jan-2018)
http://index-of.es

<1% match (student papers from 05-Sep-2017)
Submitted to University of Bradford on 2017-09-05
<1% match (Internet from 08-Sep-2008)
http://www.gi-ev.de
<1% match (Internet from 24-Jul-2014)
http://digital-forensics.sans.org
<1% match (Internet from 23-Mar-2015)
http://www.itc.nl
<1% match (Internet from 30-Apr-2012)
http://www.slideshare.net
<1% match (student papers from 18-Apr-2009)
Submitted to University of Maryland, University College on 2009-04-18
<1% match (Internet from 28-Nov-2015)
http://www.cfreds.nist.gov
<1% match (student papers from 21-Feb-2014)
Submitted to Manchester Metropolitan University on 2014-02-21
<1% match (Internet from 25-Jun-2016)
https://issuu.com/apsm/docs/asm_april_may_2016_final/35
<1% match (student papers from 16-May-2016)
Submitted to Texas A&M University, San Antonio on 2016-05-16
<1% match (student papers from 29-Oct-2012)
Submitted to University of Greenwich on 2012-10-29
<1% match (student papers from 31-Dec-2017)
Submitted to Universiti Teknologi MARA on 2017-12-31
<1% match (student papers from 14-Jul-2010)
Submitted to National University of Ireland, Maynooth on 2010-07-14
<1% match (student papers from 22-Mar-2013)
Submitted to Manchester Metropolitan University on 2013-03-22

<1% match (student papers from 11-Jan-2013)
Submitted to City University on 2013-01-11
<1% match (Internet from 15-Sep-2017)
http://dspace.mit.edu
<1% match (student papers from 18-Apr-2013)
Submitted to Strathmore University on 2013-04-18
<1% match (student papers from 12-Sep-2016)
Submitted to CTI Education Group on 2016-09-12
<1% match (Internet from 01-Sep-2011)
http://www.ogcio.gov.hk
<1% match (Internet from 28-Nov-2011)
http://essay.utwente.nl
<1% match (student papers from 14-May-2015)
Submitted to University of Greenwich on 2015-05-14
<1% match (Internet from 09-Sep-2017)
http://dspace.udel.edu
<1% match (Internet from 09-Mar-2016)
http://opus.bath.ac.uk
<1% match (Internet from 28-Mar-2018)
https://repository.up.ac.za/bitstream/handle/2263/44916/Maepa_Success_2014.pdf?sequence=
<1% match (Internet from 17-Sep-2017)
http://www.enterpriseappstoday.com
<1% match (Internet from 19-Feb-2011)
http://etd.lib.fsu.edu
<1% match (Internet from 16-Mar-2017)
http://documents.mx
<1% match (student papers from 10-Oct-2006)
Submitted to Colorado Technical University Online on 2006-10-10

<1% match (student papers from 29-Aug-2016)
Submitted to Universiti Teknikal Malaysia Melaka on 2016-08-29
<1% match (student papers from 19-Mar-2015)
Submitted to Strathmore University on 2015-03-19
<1% match (Internet from 17-Feb-2017)
https://shareok.org/bitstream/handle/11244/11966/Thesis-1998-D326r-1.pdf?isAllowed=y&sequence=1
<1% match (Internet from 19-Aug-2011)
http://mincom.com.ph
<1% match (student papers from 07-Feb-2014)
Submitted to London School of Marketing on 2014-02-07
<1% match (Internet from 21-Oct-2013)
http://www.hackerzvoice.net
<1% match (Internet from 09-Dec-2017)
http://forensicswiki.org
<1% match (student papers from 01-Jul-2012)
Submitted to Institute of Graduate Studies, UiTM on 2012-07-01
<1% match (student papers from 29-Apr-2011)
Submitted to RDI Distance Learning on 2011-04-29
<1% match (student papers from 07-Jul-2014)
Submitted to Petroleum Gas University of Ploiesti on 2014-07-07
<1% match (Internet from 13-Nov-2013)
http://thp.cc
<1% match (Internet from 20-Feb-2017)
https://espace.curtin.edu.au/bitstream/handle/20.500.11937/1836/198085_Burns%202014.pdf?isAllowed=y&sequence=2
<1% match (Internet from 20-Nov-2017)
http://etd.lib.metu.edu.tr

<1% match (Internet from 11-Mar-2016)
http://dspace.brunel.ac.uk
<1% match (Internet from 22-Oct-2015)
http://mybeibei.net
<1% match (Internet from 29-Nov-2015)
http://citeseerx.ist.psu.edu
<1% match (Internet from 15-Apr-2017)
https://www.scribd.com/document/332142909/apps-pdf
<1% match (Internet from 30-Jan-2015)
http://www.itc.nl
<1% match (Internet from 27-Mar-2014)
http://waset.org
<1% match (Internet from 12-Apr-2017)
http://media.corporate-ir.net
<1% match (Internet from 16-Sep-2017)
https://digital.library.unt.edu/ark:/67531/metadc791539/m2/1/high_res_d/861891.pdf
<1% match (Internet from 16-Feb-2017)
http://www.uta.edu
<1% match (Internet from 04-Jun-2017)
http://repository.um.edu.my
<1% match (Internet from 31-Dec-2012)
http://www.cszhe.info
<1% match (publications)
"Information Security and Assurance", Springer Nature, 2011
<1% match (student papers from 23-Jun-2011)
Submitted to Higher Education Commission Pakistan on 2011-06-23
<1% match (publications)
"Fundamentals of Secure Proxies", Security Intelligence, 2015.