SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

DESIGN AND IMPLEMENTATION OF A SYSTEM TO COLLECT POTENTIAL EVIDENCES FROM PRIVATE CLOUD PLATFORM FOR EFFECTIVE FORENSIC ANALYSIS

SUBMITTED TOWARDS THE PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING(Computer Engineering)

BY

Rohit Gund	**B120024243**
Atharv Vibhute	**B120024347**
Aditya Dhage	**B120024227**
Tushar Chopade	**B120024224**

UNDER THE GUIDANCE OF

Dr. Vrushali Kulkarni(Internal Guide) and
Prof. Prasad Purnaye(Co-Guide)

Department Of Computer Engineering
MAEERs MAHARASHTRA INSTITUTE OF TECHNOLOGY
Kothrud, Pune 411 038

2017-2018

MAHARASHTRA ACADEMY OF ENGINEERING AND EDUCATIONAL RESEARCH'S

MAHARASHTRA INSTITUTE OF TECHNOLOGY

PUNE

DEPARTMENT OF COMPUTER ENGINEERING

C E R T I F I C A T E

This is to certify that

DESIGN AND IMPLEMENTATION OF A SYSTEM TO COLLECT POTENTIAL EVIDENCES FROM PRIVATE CLOUD PLATFORM FOR EFFECTIVE FORENSIC ANALYSIS

Submitted by

Rohit Gund	**B120024243**
Atharv Vibhute	**B120024347**
Aditya Dhage	**B120024227**
Tushar Chopade	**B120024224**

is a bonafide work carried out by Students under the supervision of Dr. Vrushali Kulkarni and it is submitted towards the partial fulfilment of the requirement of Bachelor of Engineering (Computer Engineering).

Dr. Vrushali Kulkarni
Internal Guide

Dr.Vrushali Kulkarni
H.O.D.

Dr. L. K. Kshirsagar
Principal
Maharashtra Institute of Technology

PROJECT APPROVAL SHEET

A Project title

DESIGN AND IMPLEMENTATION OF A SYSTEM TO COLLECT POTENTIAL EVIDENCES FROM PRIVATE CLOUD PLATFORM FOR EFFECTIVE FORENSIC ANALYSIS

Is successfully completed by

Rohit Gund	**B120024243**
Atharv Vibhute	**B120024347**
Aditya Dhage	**B120024227**
Tushar Chopade	**B120024224**

at

DEPARTMENT OF COMPUTER ENGINEERING

MAHARASHTRA INSTITUTE OF TECHNOLOGY

SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

ACADEMIC YEAR 2017-2018

Dr. Vrushali Kulkarni	Dr. Vrushali Kulkarni
Internal Guide	H.O.D.
Dept. of Computer Engg.	Dept. of Computer Engg.

Abstract

This is research based project on cloud forensics that mainly focuses on making cloud more forensics friendly. This project has smart agent implied in client machine that live time traces some logs and other data that can be treated as potential evidences and send them to server in a encrypted form.A special program at Server side performs analytics over data collected through smart agent. To generate potential evidences and stores them in a repository . Such evidences can be made available for cyber forensics experts through a web portal.

Keywords:

Keywords:Cloud Forensics,Data Analysis, Data Encryption,Cyber Forensics

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the preliminary project report on 'Cloud Forensic'.We would like to take this opportunity to thank our internal guide Dr. Vrushali Kulkarni for giving us all the help and guidance we needed. We are really grateful for her kind support. Her valuable suggestions were very helpful.

We are also grateful to Prof. Prasad Purnaye for his indispensable support, suggestions and constant motivation as our Co - Guide..

Rohit Gund

Atharv Vibhute

Aditya Dhage

Tushar Chopade

(B.E. Computer Engineering)

Contents

List of Figures

List of Tables

Chapter 1

SYNOPSIS

1.1 Project Title

The title of our project is 'DESIGN AND IMPLEMENTATION OF A SYSTEM TO COLLECT POTENTIAL EVIDENCES FROM PRIVATE CLOUD PLATFORM FOR EFFECTIVE FORENSIC ANALYSIS'

1.2 Project Option

The option of our project is Inhouse

1.3 Internal Guide

The internal guide of our project is Dr. Vrushali Kulkarni and co-guide is Prof. Prasad Purnaye

1.4 Sponsorship and External Guide

Our project is Inhouse.

1.5 Technical Keyword

- Cloud Forensic
- Cloud service provider
- Cloud Security
- Virtual Machine
- Forensic Investigation

1.6 Problem Statement

Design And Implementation Of A System To Collect Potential Evidences From Private Cloud Platform For Effective Forensic Analysis

1.7 Abstract

This is research based project on cloud forensics that mainly focuses on making cloud more forensics friendly. This project has smart agent implied in client machine that live time traces some logs and other data that can be treated as potential evidences and send them to server in a encrypted form.A special program at Server side performs analytics over data collected through smart agent. to generate potential evidences and stores them in a repository .Such evidences are made available for cyber forensics experts through a web portal. Keywords:Cloud Forensics,Data Analysis, Data Encryption,Cyber Forensics

1.8 Goals and Objectives

1. To create a smart virtual agent who identifies potential evidence from running vms.
2. To provide Access control mechanism for evidence repository.
3. Evidences encryption for security concerns.
4. Web access portal to the evidence data.
5. To make cloud more forensic friendly.

1.9 Relevant Mathematics Associated With The Project

1.9.1 Mathematical Model

A mathematical model is a description of system using mathematical concepts and language. A model may help to explain a system and to study the effects of different components, and to make predictions about behaviour.

Mathematical models can take many forms, including but not limited to dynamical systems, statistical models, differential equations, or game theoretic models. These and other types of models can overlap, with a given model involving a variety of abstract structures. In general, mathematical models may include logical models.

In many cases, the quality of a scientific field depends on how well the mathematical models developed on the theoretical side agree with results of repeatable experiments. Lack of agreement between theoretical mathematical models and experimental measurements often leads to important advances as better theories are developed.

Mathematical Model for Cloud Forensic :

- I: Set of Inputs
- O: Set of outputs
- F: Functions
- Sc: Success cases
- Fc: Failure cases

1.9.2 Mathematical Model : Smart Agent

- I: {} where,
- O: {O1,O2} where,
  - O1= Logs
  - O2= Status.
- F: {F1,F2} where,
  - F1=RequestData() — Request Data From VM
  - F2=SendData — Send received data to Agent Manager
- Sc: {Sc1, Sc2} where,
  - Sc1. Got correct data from VM
  - Sc2. VM is live. .
- Fc: {Fc1, Fc2} where,
  - Fc1. Data not received.
  - Fc2. VM is dead.
  - Fc3. Smart agent crashed

1.9.3 Mathematical Model : Agent Manager

- I: {I1} where,
  - I1=Logs from smart agent
- O: {O1} where,
  - O1= Smart Agent Status
- F: {F1,F2} where,
  - F1=createEvidenceUnprocessedDatabase() — Create Database
  - F2=requestStatus() —Current Status of Smart Agent
- Sc: {Sc1, Sc2} where,
  - Sc1. Received correct data from smart agent
  - Sc2. Data stored according to each VM
- Fc: {Fc1, Fc2} where,
  - Fc1. Evidence Unprocessed database not got created
  - Fc2. Data not stored correctly
  - Fc3. Mixing of data from all VMs

1.9.4 Mathematical Model : Evidence Manager

- I: {I1} where,
  - I1=Raw potential evidence
- O: {O1,O2,O3} where,
  - O1 = Analysis Successful
  - O2 = Formatted Successful
  - O3 = Organized Successful
- F: {F1,F2,F3,F4,F5} where,
  - F1 = analyze() — Unprocessed Evidence DB analyzed for evidence
  - F2 = format() — Evidence Formatting
  - F3 = organize() — User wise Evidence Organization
  - F4 = createEvidenceDatabase() — Creating DB and Store Potential evidence
  - F5 = grantAccess() — Giving access to Forensic Expert
- Sc: {Sc1, Sc2} where,
  - Sc1. Data analyzed, format, organized correctly
  - Sc2. Data stored in evidence database
- Fc: {Fc1, Fc2} where,
  - Fc1. Mixing of data
  - F2. Incorect database access

1.9.5 Mathematical Model : Evidence Access Portal

- I: {I1} where,
  - I1=Login Credentials
- O: {O1,O2} where,
  - O1= Access granted
  - O2= Potential Evidence
- F: {F1,F2} where,
  - F1 = login() — Forensic Expert Login Authentication
  - F2 = requestData() — Forensic Expert Requesting data from evidence DB
  - F3 = displayData() — Potential Evidence display on Portal
  - F4 = addLog() — Forensic Experts login logs stored
- Sc: {Sc1, Sc2} where,
  - Sc1. Access Granted
  - Sc2. Logs created for each access
- Fc: {Fc1, Fc2} where,
  - Fc1. Incorrect evidence displayed
  - Fc2. Login authentication failed
  - Fc3. Incorrect Database access

1.10 Names of Conferences / Journals Where Papers Can Be Published

- IEEE/ACM Conference/Journal 1
- Conferences/workshops in IITs
- Central Universities or SPPU Conferences
- IEEE/ACM Conference/Journal 2

1.11 Plan of Project Execution

Tasks	June	July	Aug	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Group Formation and Finallization	■											
Topic Search		■										
Preliminary Information Gathering		■										
Discussion with project coordinator and topic finalization		■										
Synopsis Preparation and submission			■									
Detailed Literature Survey				■								
Preparing interim report					■							
Language Study						■						
Open Nebula and forensics tools study				■								
Coding and Implementation							■	■	■	■		
Testing									■	■	■	
Final Documentation											■	
Final Report Submission												■

Figure 1.1: Execution Plan

Chapter 2

TECHNICAL KEYWORDS

2.1 Area of Project

The area of our project is Cloud Forensics .

2.2 Technical Keywords

A. Hardware

- Ubuntu OS
- 4GB RAM or more

B. Architecture

- Client Server Architecture
- Real time Database Connectivity

C. Networking

- Server

Chapter 3

INTRODUCTION

3.1 Project Idea

The overall Idea of the project is to make cloud more forensic friendly.We are going to create smart virtual agent in deployed virtual machine. So that we can trace all the ongoing activity and collect potential evidence. Due to this it is easy for forensic investigator to use that evidence to solve crime.

3.2 Motivation of the Project

Rapid advancements and increase in popularity in cloud technology is certainly pushing cloud forensic to new level.The existing cloud system does not keep track of what user is doing on the provided systems.Even if any attack is done using the cloud systems no evidences can be traced back.So for eliminating the lack of monitoring capabilities in current system we have come up with a solution where we can design asystem that will trace as well as monitor important things aboutcustomers usage .So when any attack on any system takes placewe can generate potential evidence helpingthe entire forensicsprocedure.

3.3 Literature Survey

Table 3.1: Literature Survey

Sr.No	Title of Paper	Year of Publication	Author	Findings
1.	Cloud Forensics: A Meta-Study of Challenges, Approaches, and Open Problems	2014	Shams Zawoad , Ragib Hasan	Actual challenges in cloud forensics,And approach to find solution
2.	RightScale 2017 State of Cloud Report	2017	RightScale	Survey of work done in cloud forensics,scope of work in cloud forensics, statastics of Cloud
3.	Towards a security-enhanced PaaS platform for multi-cloud applications	2016	Kyriakos Kritikos, Tom Kirkham, Bartosz Kryza, Philippe Massonet	Security Concerns and solution for PasS model for multi Cloud ,Forensics status in multi cloud platform
4.	NIST Cloud Computing Forensic Science Challenges	2014	NIST Cloud Computing Forensic Science Working Group Information Technology Laboratory	Challenges in cloud like evidence data gathering and data integrity
5.	Towards Transparent and trustworthy cloud	2016	Marco Anisetti, Claudio A. Ardagna, and Ernesto Damiani,	Cloud providers have deployed security controls to prevent attacks and unauthorized activities.

Sr.No	Title of Paper	Year of Publication	Author	Findings
6.	Cloud-Trust - a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds	2014	Dan Gonzales, Member, IEEE, Jeremy Kaplan, Evan Saltzman, Zev Winkelman, Dulani Woods	Cloud architecture reference model that incorporates a wide range of security controls and that estimates high level security metrics to quantify the degree of confidentiality and integrity offered by a CCS or cloud service provider (CSP).
7.	Guide to Integrating Forensic Techniques into Incident Response	2006	Karen Kent, Suzanne Chevalier, Tim Grance, Hung Dang	Collecting, Examining and Analysis of evidence data.
8.	Cloud Log Forensics: Foundations, State of the Art and Future Directions	2016	Suleman Khan, Abdullah Gani, Ainuddin Wahid, Abdul Wahab, Mustapha Aminu Bagiwa, Muhammad Shiraz, Samee U. Khan, Rajkumar Buyya and Albert Y. Zomaya	Review the state of the art of CLF and highlights different challenges and issues involved in investigating cloud log data.
9.	Log Management in the Cloud: A Comparison of In-House versus Cloud-Based Management of Log Data	2008	Jerry Shenk	Detailed information about log management in cloud systems and compare two type of log management systems available.

Chapter 4

PROBLEM DEFINITION AND SCOPE

4.1 Problem Statement

Design And Implementation Of A System To Collect Potential Evidences From Private Cloud Platform For Effective Forensic Analysis

4.1.1 Goals and Objectives

1. To create a smart virtual agent who identifies potential evidence from running VMs.
2. To provide Access control mechanism for evidence repository.
3. Evidences encryption for security concerns.
4. Web access portal to the evidence data.
5. To make cloud more forensic friendly.

4.1.2 Statement of Scope

We describe what features are in the scope of the software and what are not in scope of the software to be developed.

Initially we are limiting the project to the Paas model of the cloud systems. We are going to develop a system that will keep track of potential evidence from systems deployed to the user and will generate some evidences from logs it will maintain and store them in a database. Later the database can be accessed by Forensics experts. We will also try to maintain a trace of who is using that evidence database so that we can maintain the user's privacy. Initially This System will work only with Platform As A Service(PaaS) Model of Cloud that will try to track user activities.

4.2 Major Constraints

1. The Operting system used in Client side should be Ubuntu OS.
2. The client side OS should have Mongo database(version 3.6 or above) and Pymongo.
3. Php version should be 5.6 for web portal.

4.3 Methodologies Of Problem Solving And Efficiency Issues

A problem can be solved by many different solutions. The methodologies of problem solving considers the performance parameters for each approach. Thus considering the efficiency issues.

1. Our project is based on Linux operating system but it can be implemented on other operating systems as well. However we have selected Linux operating because it is much more popular in comparison to other OS which increases the usability of our application.Also Most of IT companies uses Linux so we choose linux for our project.

4.4 Outcome

Forensic expert can use Evidence Access Portal to examine cases and prove or disprove cases using collected evidences.

4.5 Hardware Resources Required

1. 1TB Hard disk
2. 4GB RAM minimum,8GB RAM recommended
3. Intel Core i3 Processors
4. Computer with ubuntu 16.04 OS

4.6 Software Resources Required

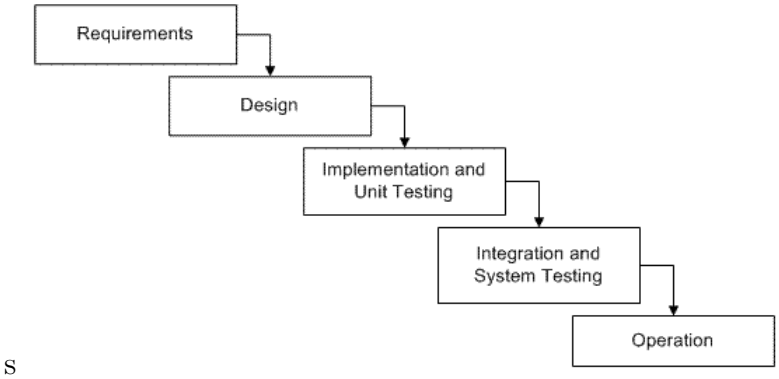
- OpenNebula 5.4
- Operating System: Ubuntu 16.04
- Programming Language: Java, Python, PHP, HTML, CSS

Chapter 5

PROJECT PLAN

5.1 Project Estimates

In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.



s

Figure 5.1: Waterfall model

The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:**

  All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

  - **Hardware Requirements:**
    - 1TB Hard disk
    - 4GB RAM minimum,8GB RAM recommended
    - Intel Core i3 Processors
    - Computer with ubuntu 16.04 OS
  - **Software Requirements:**
    - OpenNebula 5.4
    - Operating System: Ubuntu 16.04
    - Programming Language: Java, Python, PHP, HTML, CSS

- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. The following units were serially developed:

  1. Smart Agent
  2. Agent Manager
  3. Evidence Manager
  4. Evidence Access Portal

- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.

- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

  Here, if needed by the developer, he can extend the application usage to third party apps.

Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short

5.1.1 Reconciled Estimates

- **Cost Estimate**

  Cost Estimate: Free of cost.

- **Time Estimate**

  Time Estimates: 8 months

5.1.2 Project Resources

- **Hardware Resources:**
  - 1TB Hard disk
  - 4GB RAM minimum,8GB RAM recommended
  - Intel Core i3 Processors
  - Computer with ubuntu 16.04 OS
- **Software Resources:**

  - OpenNebula 5.4
  - Operating System: Ubuntu 16.04
  - Programming Language: Java, Python, PHP, HTML, CSS

- **Human Resources:**

  The team includes four members who are involved in making the project. Our internal guides helped to solve our doubts and remove the errors in the project.

5.2 Risk Management W.R.T. Np Hard Analysis

5.2.1 Risk Identification

Risk identification is the process of determining risks that could potentially prevent the program, enterprise, or investment from achieving its objectives. It includes documenting and communicating the concern.

For risks identification, review of scope document, requirements specifications and schedule is done. Answers to questionnaire revealed some risks. Each risk is categorized as per the categories.
The following questions are asked for identification of various categorical risks:

1. Have top software and customer managers formally committed to sup- port the project?
2. Are end-users enthusiastically committed to the project and the sys- tem/product to be built?
3. Are requirements fully understood by the software engineering team and its customers?
4. Have customers been involved fully in the definition of requirements?

5. Do end-users have realistic expectations?

6. Does the software engineering team have the right mix of skills?

7. Are project requirements stable?

8. Is the number of people on the project team adequate to do the job?

9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

5.2.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

ID	Risk Description	Probability	Impact	
			Quality	Overall
1	Top software and customer managers formally committed to support the project	High	Low	Low
2	End-users enthusiastically committed to the project and the system/product to be built	High	Low	Low
3	Requirements fully understood by the software engineering team and its customers	High	Low	Low
4	Customers been involved fully in the definition of requirements	High	Low	Low
5	End-users‘ realistic expectations	High	Low	Low
6	The software engineering team have the right mix of skills	High	Low	Low
7	Project requirements stablility	High	Low	Low
8	Number of people on the project team adequate to do the job	High	Low	Low
9	All customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built	High	Low	Low

Table 5.1: Risk Table

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 5.2: Risk Probability definitions

Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 5.3: Risk Impact definitions

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

Risk ID	1
Risk Description	Top software and customer managers formally committed to support the project
Category	Development Environment.
Source	Platform: OpenNebula 5.4, Operating System: Ubuntu 16.04, Programming Language: Java, PHP, Python, HTML, CSS
Probability	High
Impact	Low
Response	Aggravate
Strategy	Discussion of plan & steps for implementation & improvement
Risk Status	Not occurred

Risk ID	3
Risk Description	Requirements fully understood by the software engineering team and its customers
Category	Requirements
Source	Platform: OpenNebula 5.4, Operating System: Ubuntu 16.04, Programming Language: Java, PHP, Python, HTML, CSS
Probability	High
Impact	Low
Response	Aggravate
Strategy	Proceed with the implementation
Risk Status	Identified

Risk ID	5
Risk Description	End-users‘ realistic expectations
Category	Customers‘ Requirements
Source	This was identified during early development and testing.
Probability	High
Impact	Low
Response	Accept
Strategy	Advance as per users‘ need & make necessary changes
Risk Status	Identified

5.3 Project Schedule

5.3.1 Project task set

Major Tasks in the Project stages are:

- Task 1: Selecting the project domain & the project.
- Task 2: Checking for duplicity & research work.
- Task 3: Getting the company’s sponsorship.
- Task 4: Communication with the internal & external guide.
- Task 5: Project Planning.
- Task 6: Study of tools & technology- Android Studio, Bluetooth LE.

- Task 7: Design Phase.
- Task 8: Implementation & Coding.
- Task 9: Testing.
- Task 10: Deployment.
- Task 11: Final Presentation.

5.3.2 Timeline Chart

See Table below

Table 5.4: Plan of Project

From	To	Task	Status
27–06–2017	30–06–2017	Group Formation and finalization	Done
01–07–2017	15–07–2017	Topic Search	Done
16–07–2017	20–07–2017	Preliminary Information Gathering	Done
21–07–2017	28–07–2017	Project Dicsussion with Project Coordinator and topic finalization	Done
01–08–2017	04–08–2017	Synopsis preparation and submission	Done
25–08–2017 19–09–2017	30–08–2017 24–09–2017	Detailed Literature Survey .	Done .
25–09–2017	06–10–2017	Preparing Interim report	Done
20–12–2017	02–01–2018	Language Study	Done
03–01–2018	20–01–2018	Forensics Process analysis and Study	Done
21–01–2018	04–03–2018	Coding and Implementation	Done
05–03–2018	21–04–2018	Testing	Done
13–04–2018	21–04–2018	Final Documentation	Done
25–04–2018	20–05–2018	Final Project Report	Done

5.4 Team Organization

The team structure is a newer, less hierarchical organizational structure in which individuals are grouped into teams.

A team should be a group of workers, with complementary skills and synergistic efforts, all working toward a common goal. An organization may have several teams that can change over time. Teams that include members from different functions are known as cross-functional teams.

5.4.1 Team structure

We have completed the journal, report and partial implemention of our project. The project was equally divided and distributed among the team members.Under the guidance of our external guide, we have implemented various adaptations in our project.

5.4.2 Management reporting and communication

Emails, phone calls and messages were used to communicate within the team. The team met in person as well to do different tasks of the project. We also communicated with our external guide via emails and messages.In cases where we required assistance in our project. We met our guide who helped to remove our difficulties and solve our doubts.

Chapter 6

SOFTWARE REQUIREMENT SPECIFICATION

6.1 Introduction

6.1.1 Purpose and Scope of Document

- The reader can understand basic functionality of the project
- Also he can understand the working and architectural setup of the project

6.1.2 Overview of responsibilities of Developer

- Coding
- Implementation
- Testing

6.2 Usage Scenarios

A usage scenario, or scenario for short, describes a real-world example of how one or more people or organizations interact with a system. They describe the steps, events, and/or actions which occur during the interaction. Usage scenarios can be very detailed, indicating exactly how someone works with the user interface, or reasonably high-level describing the critical business actions but not indicating how they're performed. The basic strategy is to identify a path though a use case, or through a portion of a use case, and then write the scenario as an instance of that path.

Scenario:

1. Access web portal
2. Login using credentials.
3. Use potential evidences to prove or disprove case.

6.2.1 User profiles

- Developer :
  - They create the platform and analyse customer feedbacks.
  - They should be knowledgeable about end users.
- Forensic Expert :
  - They should have valid login credentials.
  - They should be knowledgeable about the platform.

6.2.2 Use cases

Sr. No.	Use Cases	Description	Actors	Assumptions
1	Use case 1	Cloud Forensics	VM, Smart Agent, Agent Manager, Evidence Manager, Evidence Access Portal	Server is switched on

Table 6.1: Use Cases

6.2.3 Use case view

Use case diagram shows the overall functionality of the system. In our system there are three use case diagrams.

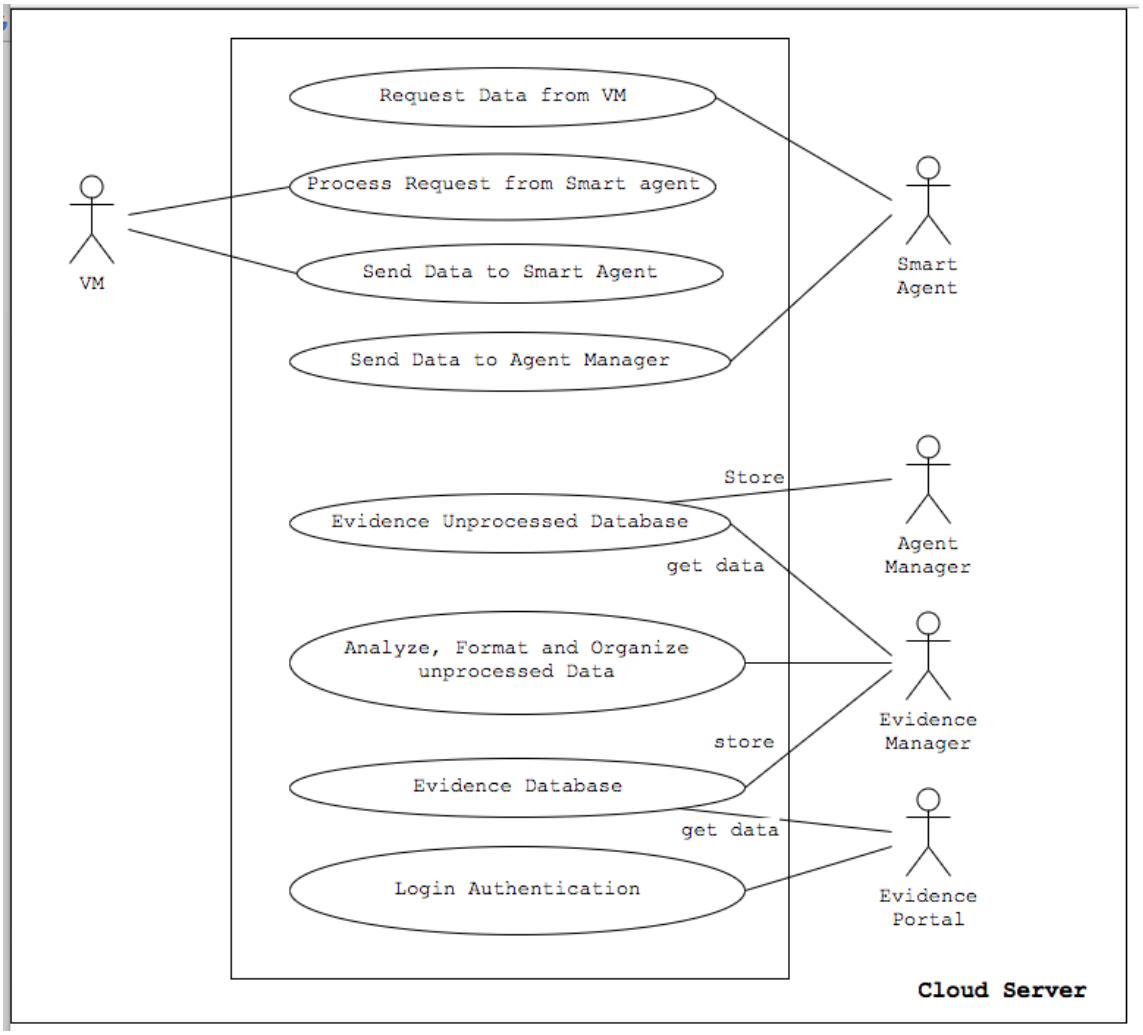


Figure 6.1: Use Case diagram

6.3 UML modeling

6.3.1 Activity diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows).

6.3.2 Component diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

6.3.3 Deployment diagram

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process.

6.3.4 Sequence diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

6.3.5 Statechart diagram

A state diagram, also called a state machine diagram or statechart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language (UML).

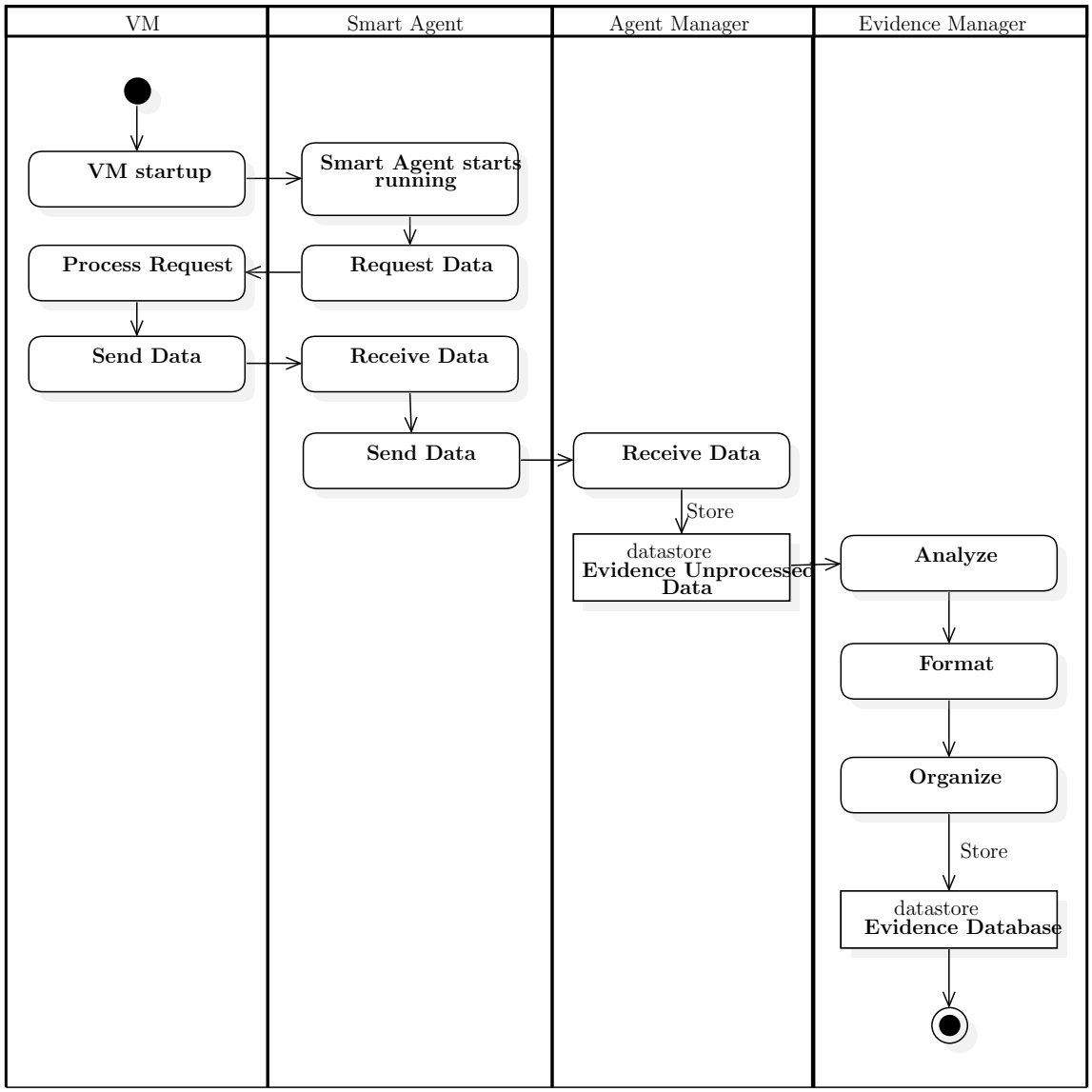


Figure 6.2: Activity Diagram

6.4 Functional Model And Description

1. **Smart Agent**

   Smart agent collects the logs. It will monitor running processes and files updated by them. It will check recently updated files and send them to agent manager. Data will be encrypted before sending for security concerns. Then encrypted data will be send to Agent Manager using SSH. Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. The best-known example application is for remote login to computer systems by users. SSH provides a secure channel over an unsecured network in a client server architecture, connecting an SSH client application with an SSH server.

2. **Agent Manager**

   Agent manager communicates with Smart Agent via SSH. It is responsible for decryption of data received from Smart Agent. It creates evidence unprocessed database and stores data received from Smart Agent. It will monitor All VM and keeps track of Smart Agent Availability.

3. **Evidence Manager**

   Evidence Manager is responsible for analysis of unprocessed evidences. Data will be formatted into relevant format and user wise organized. Finally, it will store potential evidences into evidence database.

4. **Evidence Access Portal**

   Forensic experts will get access to potential evidences by using Evidence Access Portal. Experts authentication logs are maintained for chain-of- custody and data integrity.

6.4.1 Non Functional Requirements

- **Interface Requirements:**

  Your tool's user interface is everything that the user can see and interact with. There are many frameworks available for HTML and CSS to design responsive portals.

- **Performance Requirements:**

  - Choosing the right algorithms and data structures should always be the priority.
  - Do not allocate memory if you can avoid it
  - To ensure your portal performs well across a wide variety of devices, ensure your code is efficient at all levels and aggressively optimize your performance.

  - **Avoid Creating Unnecessary Objects**

    Object creation is never free. A generational garbage collector with per-thread allocation pools for temporary objects can make allocation cheaper, but allocating memory is always more expensive than not allocating memory.

  - **Prefer Static Over Virtual**

    If you do not need to access an object's fields, make your method static. Invocations will be about 15%-20% faster. It's also good practice, because you can tell from the method signature that calling the method can't alter the object's state.

  - **Consider Package Instead of Private Access with Private Inner Classes**

- **Software quality attributes:**
  - Availability
    - It is the proportion of time a system is in a functioning condition.
    - Availability of a system is typically measured as a factor of its reliability as reliability increases, so does availability.
    - Reliability needs to be evaluated and improved related to both availability and the cost of ownership (due to cost of spare parts, maintenance man-hours, transport costs, storage cost, part obsolete risks etc.).
    - Fault tree analysis and related software are developed to calculate (analytic or by simulation) availability of a system or a functional failure condition within a system.
  - Modifiability
    - Portability
      - Portability in high-level computer programming is the usability of the same software in different environments. The prerequirement for portability is the generalized abstraction between the application logic and system interfaces.
      - Our application is portable.
    - Reusability
      - In computer science and software engineering, reusability is the use of existing assets in some form within the software product development process. Assets are products and by-products of the software development life cycle and include code, software components, test suites, designs and documentation.

- Subroutines or functions are the simplest form of reuse. A chunk of code is regularly organized using modules or namespaces into layers.
- The vibration module in our code has been re-used for various types of incoming notifications.

* Scalability
  - An algorithm, design, networking protocol, program, or other system is said to scale if it is suitably efficient and practical when applied to large situations (e.g. a large input data set, a large number of outputs or users, or a large number of participating nodes in the case of a distributed system).

– Testability

* Software testability is the degree to which a software artifact (i.e. a software system, software module, requirements- or design document) supports testing in a given test context.
* It is an extrinsic property which results from interdependency of the software to be tested and the test goals, test methods used, and test resources (i.e., the test context).

– Usability

* In Software engineering, usability is the degree to which a software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.
* It includes methods of measuring usability, such as needs analysis and the study of the principles behind an object's perceived efficiency or elegance.
* The primary notion of usability is:

1. More efficient to use, takes less time to accomplish a particular task
2. Easier to learnoperation can be learned by observing the object
3. More satisfying to use

6.4.2 Software Interface Description

- We are using php,html and css to provide an interface to forensic expert.
- The software interface (Web Portal)provides various options and buttons such as login, chain of custody and functionality to display table dynamically sorted using VM name and user of VM.

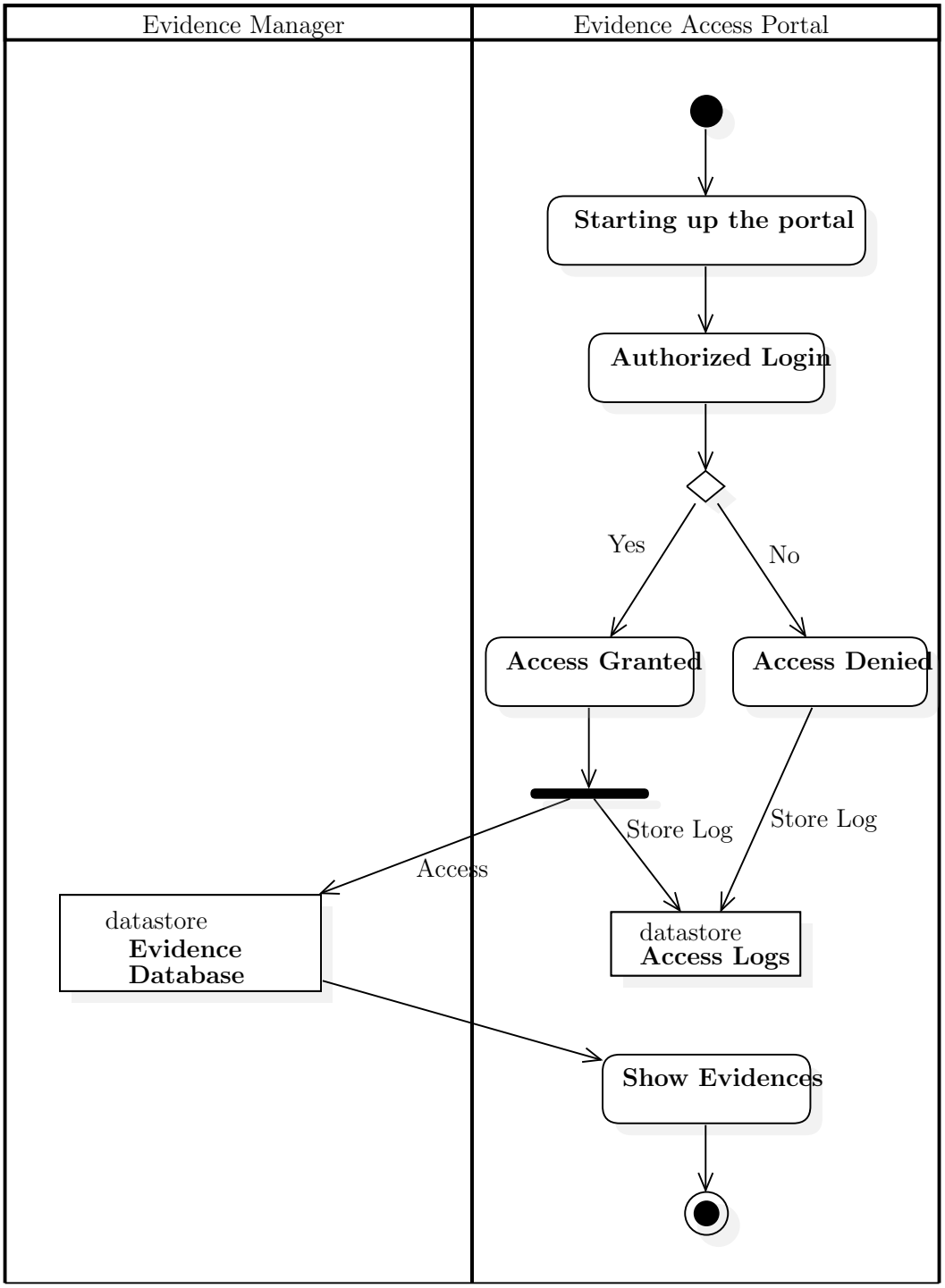


Figure 6.3: Activity Diagram of Evidence Access Portal

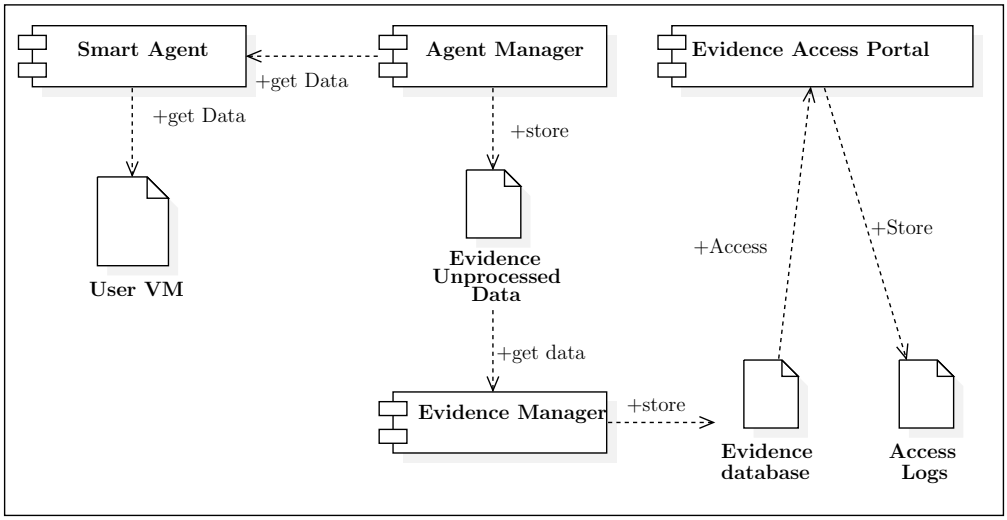


Figure 6.4: Component Diagram

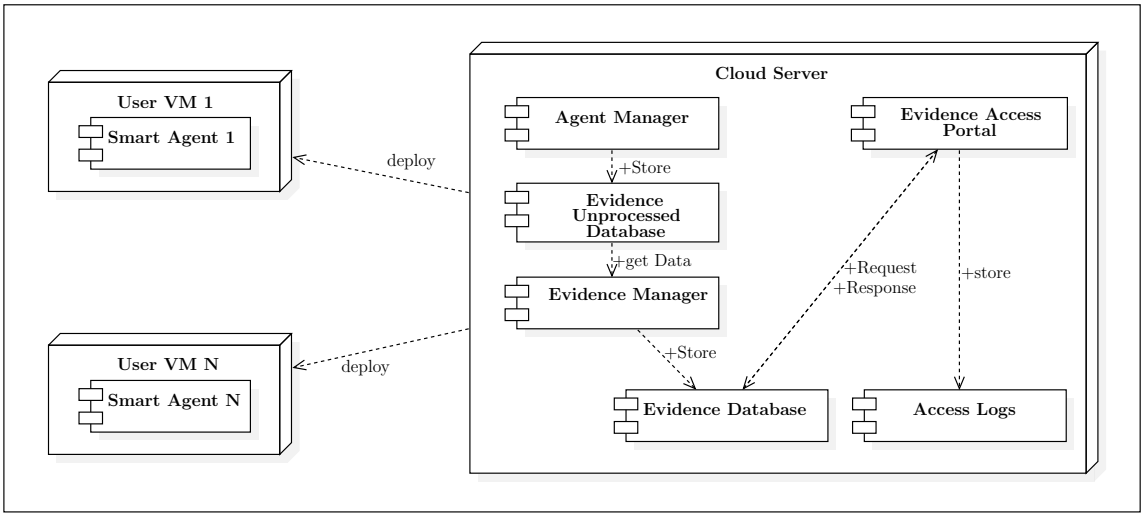


Figure 6.5: Deployment Diagram

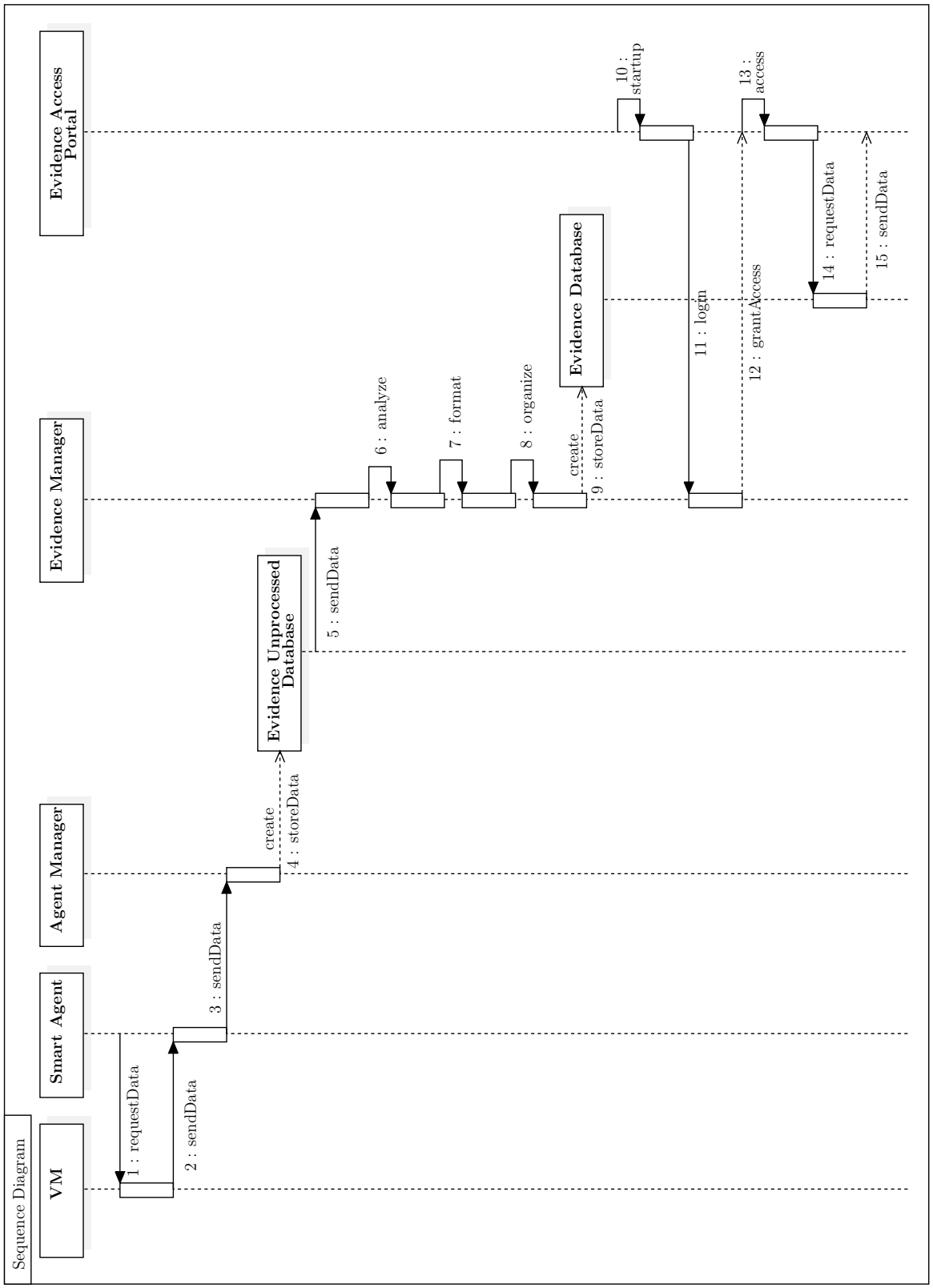


Figure 6.6: Sequence Diagram

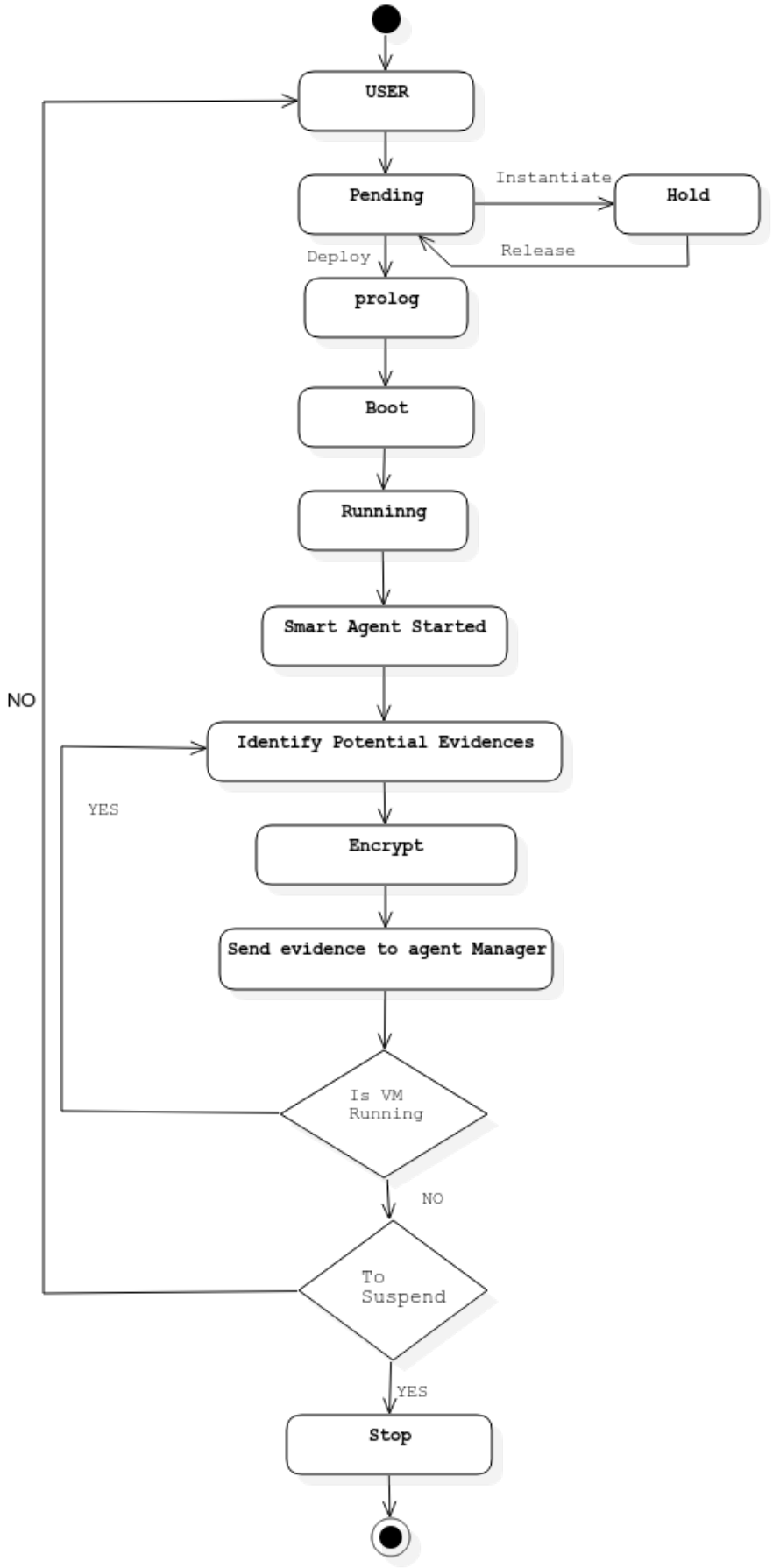


Figure 6.7: VM Agent State Diagram

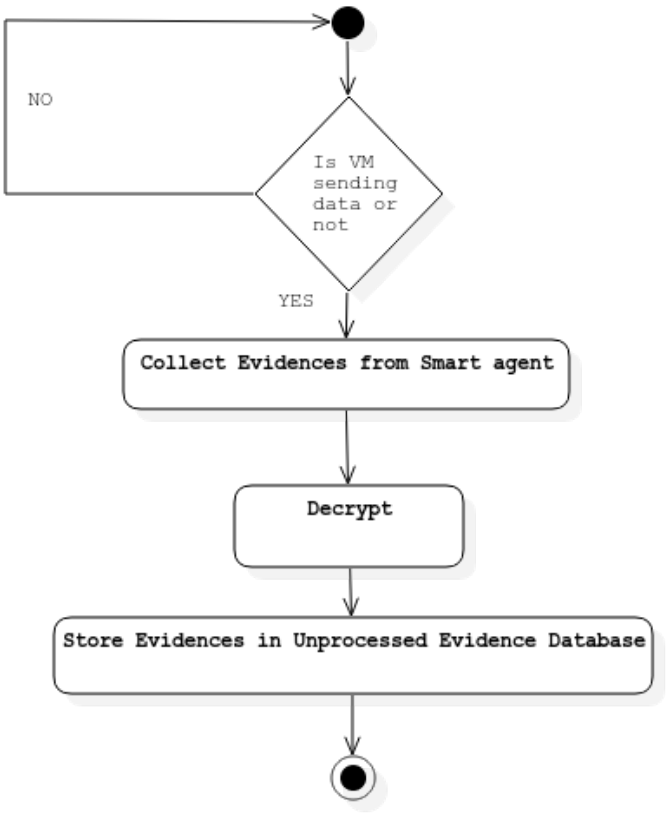


Figure 6.8: Agent Manager

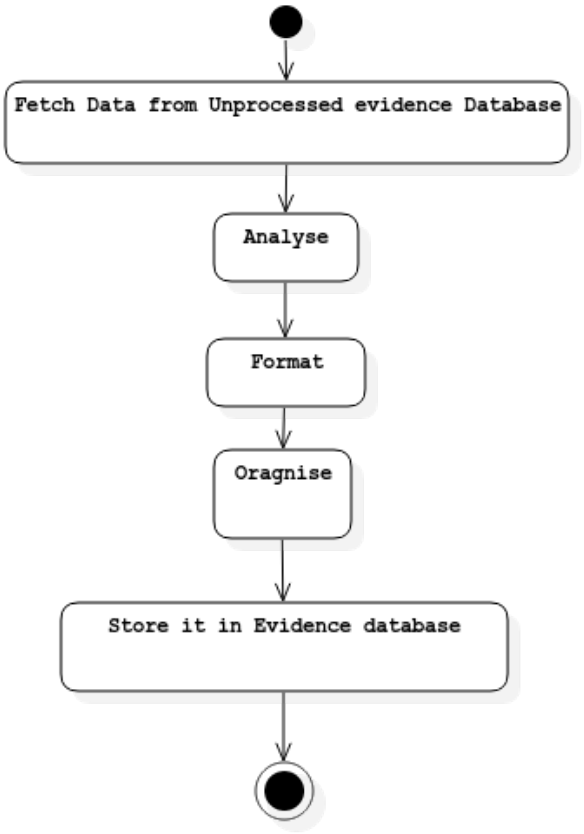


Figure 6.9: Evidence Manager

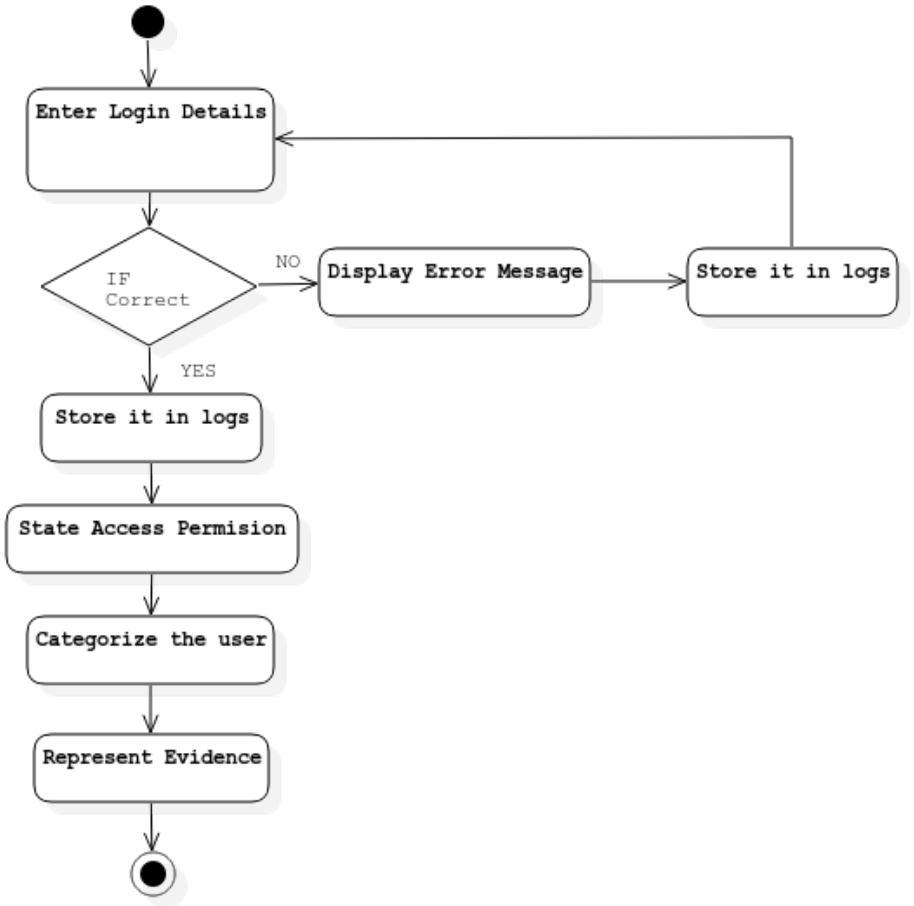


Figure 6.10: Evidence Access Portal

Chapter 7

DETAILED DESIGN DOCUMENT USING APPENDIX A AND B

7.1 Introduction

7.1.1 Purpose

The purpose of this document is to describe the implementation of the project. This solution helps forensic experts to examine cases using collected evidences.

7.1.2 Scope

This document describes the implementation details of the project. It consists of major funcions such as:

- Smart Agent
- Agent Manager
- Evidence Manager
- Evidence Access Portal

7.1.3 Definitions, Acronyms, Abbreviations

Abbreviations

- CSP: Cloud Service Provider
- COC: Chain Of Custodoy
- SLA: Service Level Agreement

Definitions

- COC: Chain of custody (CoC), in legal contexts, refers to the chronological documentation or paper trail that records the sequence of custody, control, transfer, analysis, and disposition of physical or electronic evidence.
- OpenNebula: OpenNebula provides the most simple but feature-rich and flexible solution for the comprehensive management of virtualized data centers to enable

private, public and hybrid IaaS clouds. OpenNebula interoperability makes cloud an evolution by leveraging existing IT assets, protecting your investments, and avoiding vendor lock-in.

OpenNebula is a turnkey enterprise-ready solution that includes all the features needed to provide an on-premises (private) cloud offering, and to offer public cloud services.

- SLS: A service level agreement (SLA) is a contract between a service provider (either internal or external) and the end user that defines the level of service expected from the service provider. SLAs are output-based in that their purpose is specifically to define what the customer will receive.

7.2 Architectural Design

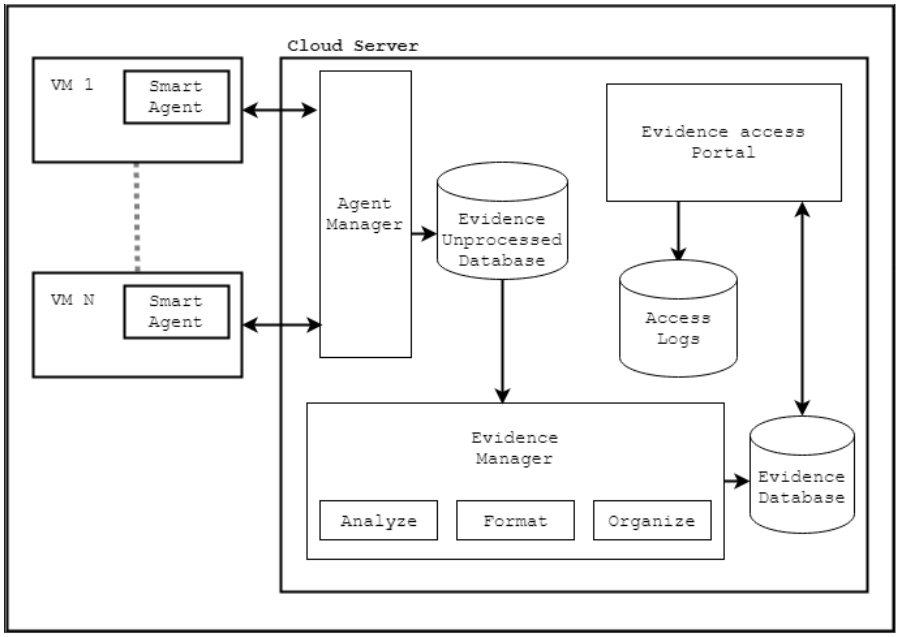


Figure 7.1: System architecture

7.2.1 Smart Agent

Smart agent collects the logs. It will monitor running processes and files updated by them. It will check recently updated files and send them to agent manager. Data will be encrypted before sending for security concerns. Encryption done by using SHA algorithm. Then encrypted data will be send to Agent Manager using SSH. Secure Shell(SSH) is acryptographic network protocol for operating network services securely over an unsecured network.The best-known example application is for remotelogginto computer systems by users. SSH provides asecure channelover an unsecured network in aclient server architecture, connecting anSSH clientapplication with anSSH server.

7.2.2 Agent Manager

Agent manager communicates with Smart Agent via SSH. It is responsible for decryption of data received from Smart Agent. It creates evidence unprocessed database and stores data received from Smart Agent. It will monitor All VM and keeps track of Smart Agent Availability.

7.2.3 Evidence Manager

Evidence Manager is responsible for analysis of unprocessed evidences. Data will be formatted into relevant format and user wise organized. Finally, it will store potential evidences into evidence database.

7.2.4 Evidence Access Portal

Forensic experts will get access to potential evidences by using Evidence Access Portal. Experts authentication logs are maintained for chain-of-custody and data integrity

7.3 Component Design

- Component-level design occurs after the first iteration of the architectural design
- It strives to create a design model from the analysis and architectural models
  - The translation can open the door to subtle errors that are difficult to find and correct later
  - "Effective programmers should not waste their time debugging they should not introduce bugs to start with." Edsgar Dijkstra
- A component-level design can be represented using some intermediate representation (e.g. graphical, tabular, or text-based) that can be translated into source code
- The design of data structures, interfaces, and algorithms should conform to well-established guidelines to help us avoid the introduction of errors

Chapter 8

PROJECT IMPLEMENTATION

8.1 Introduction

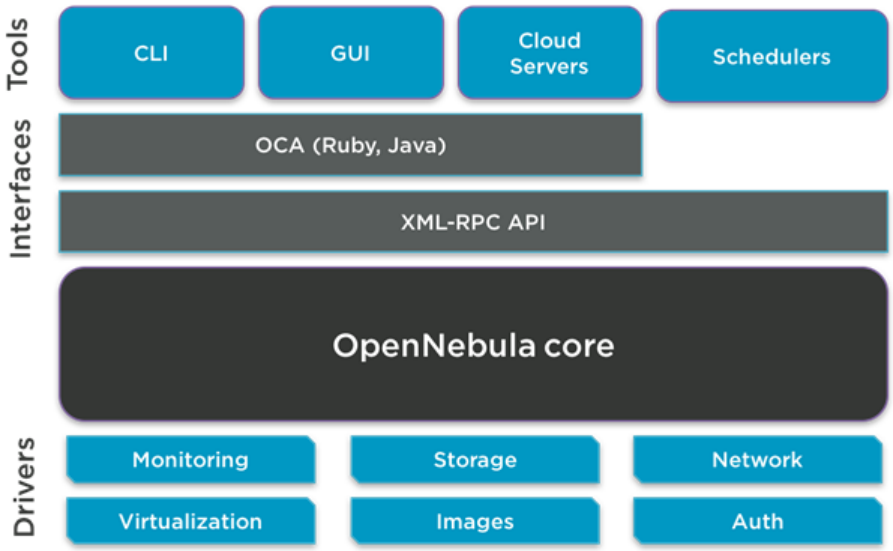
The following chapter explains the implementation of our system. We rst begin with the environment setup and installation of required software.

8.2 OpenNebula

OpenNebula provides a simple but feature-rich and exible solution for the comprehensive management of virtualized data centers to enable on-premise enterprise clouds in existing infrastructures. OpenNebula can be primarily used as a virtualization tool to manage your virtual infrastructure in the data-center or cluster, which is usually referred as Private Cloud. OpenNebula supports Hybrid Cloud to combine local infrastructure with public cloud-based infrastructure, enabling highly scalable hosting environments. OpenNebula also supports Public Clouds by providing Cloud interfaces to expose its functionality for virtual machine, storage and network management.

8.3 Overview

Figure shows the different layers of the OpenNebula architecture - Tools, Interfaces, OpenNebula Core and Drivers.



8.4 OpenNebula Features

The base setup of our entire cloud infrastructure is done on OpenNebula. We chose OpenNebula for a variety of reasons. We needed a platform that would over us more in terms of exibility, simplicity and scalability which are all features OpenNebula provides. OpenNebula is the alternative to OpenStack, which is fragmented, immature and too complex, and VMware, which is too expensive and in exible. The main features of OpenNebula are as follows :

- **Powerful and Innovative:** Most advanced and innovative enterprise-class functionality for the management of virtualized data centers to build private, public and hybrid clouds.
- **Infrastructure Agnostic:** Fully platform independent with broad support for commodity and enterprise-grade hypervisor, storage and networking resources,allowing to leverage existing IT infrastructure, protecting your investments, and avoiding vendor lock-in.
- **Adaptable, Extensible and Integrable:**Open, adaptable and extensible architecture, interfaces and components to build your customized cloud service or product.
- **Interoperable:**Cloud interoperability and portability providing cloud consumers with choice across standards and most popular cloud interfaces.
- **Fully Open Source:**OpenNebula is not a feature or performance limited edition of an Enterprise version, OpenNebula is truly open-source code, not open core, distributed under Apache license.
- **Very Light Solution:**Despite its technical sophistication and advanced functionality, OpenNebula is easy to download, install and update.

- **Stable and Proven:** Rigorously tested through an internal quality assurance process and by a large community with scalability, reliability and performance tested on many massive scalable production deployments.

- **Mature:** Development driven by user needs and matured through many release cycles.

- **Enterprise-class Product:**OpenNebula comprises all key functionalities for enterprise cloud computing, storage and networking in a single install, and ensures its long term stability and performance through a single integrated patching and updating process.

- **One-stop Support:**Wide variety of community and commercial support from the developers of OpenNebula. OpenNebula brings a rich and exible management model and the latest inno- vations in data center virtualization for the deployment of enterprise clouds in your existing infrastructure:

- **Cloud Bursting:**Extension of the local private infrastructure with resources from remote clouds.

- **On-demand Provision of Virtual Data Centers:** A Virtual Data Centers (VDC) is a fully-isolated virtual infrastructure environment where a group of users, under the control of the VDC administrator, can create and manage compute, storage and networking capacity.

- **Multiple Zones:**Centralized management of multiple instances of OpenNebula (zones) for scalability, isolation or multiple-site support.

- **Multi-VM Application Management:**Automatic execution of multi-tiered applications with auto-scaling.

8.5 Comparison with OpenStack

Comparing with Openstack, OpenNebula is an open-source effort focused on user needs, OpenStack is a vendor-driven effort. Thus, the basic reasons for choosing OpenNebula over OpenStack are as follows :

- **OpenNebula is User-driven:**OpenNebula is made for users by users, OpenStack is made for vendors by vendors. While OpenNebula is an open-source effort focused on user needs, OpenStack is a vendor-driven effort. OpenNebulas roadmap is completely driven by users needs with features that meet real demands. OpenStacks roadmap is the result from an agreement between IT vendors planning to create their own proprietary cloud solution.

- **OpenNebula is a Turnkey Solution:**OpenNebula provides the functionality offered by OpenStack Nova, Glance, Horizon, Neutron, Heat, Ceilometer and Keystone, within a single integrated enterprise-ready package. OpenNebula is delivered as a single production-proven, packaged product that comprises all key functionalities for cloud computing, storage and networking with a single install. This reduces cost and complexity of the cloud, and ensures its long term stability and performance through a single integrated patching and updating process, and one-stop support. On the other hand, OpenStack provides a set of technologies with different maturity levels that require complex integration to achieve a functional cloud infrastructure. A growing number of components and sub-projects is making even more diffcult their integration and coordination, and the delivery of a single coherent solution.

- **OpenNebula is Open-source Enterprise-ready:**There exists a single fully-open enterprise-ready OpenNebula distribution and it is not a limited edition of any enterprise version. On the other hand, the community version of OpenStack is not enterprise-ready. Any organization interested in using OpenStack,

and requiring commercial support and enterprise maturity, is recommended (by the vendors running the project) to deploy any of the many enterprise distributions. These enterprise-grade distributions incorporate different versions of the OpenStack components with extended features, custom enhancements and integrations that erode their compatibility and interoperability. Moreover they include proprietary components and exhibit significant differences in the implementation of critical underlying functionality. So the organization that chooses OpenStack is actually using proprietary software based on OpenStack, and is locked into that specific distribution given that the vendor only supports its own stack, not the community version. Even worse, there is no way to mi- grate to another vendor distribution. In other words, these distributions do not over the main benefits of open-source: low-cost, no lock-in, exibility and interoperability.

- **OpenNebula Brings Unique Features for the Enterprise Cloud:**OpenNebula brings a rich and exible management model and the latest innovations in data center virtualization for the deployment of enterprise clouds in your existing infrastructure: Cloud Bursting, on-demand provision of virtual data centers, multiple Zones, multi-VM application management with auto-scaling.

8.6 OpenNebula Installation

OpenNebula is an open-source management platform to build IaaS private, public and hybrid clouds. Installing a cloud from scratch could be a complex process, in the sense that many components and concepts are involved. The degree of familiarity with these concepts (system administration, infrastructure planning, virtualization management...) will determine the difficulty of the installation process.

8.6.1 Front-end Installation

- **Add OpenNebula Repositories:** To add OpenNebula repository on Debian/Ubuntu execute as root:

  #wget -q -O- https://downloads.opennebula.org/repo/repo.key — apt-key add -

  Ubuntu 16.04

  #echo "deb https://downloads.opennebula.org/repo/5.4/Ubuntu/16.04 stable opennebula" $ $/*etc/apt/sources.list.d/opennebula.list*

- **Installing the Software Debian/Ubuntu:** To install OpenNebula on a Debian/Ubuntu Front-end using packages from our repositories execute as root:

  #apt-get update

  #apt-get install opennebula opennebula-sunstone opennebula-gate opennebula-flow

- **Starting OpenNebula:** Log in as the oneadmin user follow these steps: The /var/lib/one/.one/one-auth fill will have been created with a randomly-generated password. It should contain the following: oneadmin:password. Feel free to change the password before starting OpenNebula. For example

  $echo "oneadmin:mypassword" > /.one/one-auth

  You are ready to start the OpenNebula daemons. You can use systemctl for Linux distributions which have adopted systemd:

  # systemctl start opennebula

  # systemctl start opennebula-sunstone

- **Verifying the Installation:** After OpenNebula is started for the first time, you should check that the commands can connect to the OpenNebula daemon. You can do this in the Linux CLI or in the graphical user interface: Sunstone.Linux CLI In the Front-end, run the following command as oneadmin:

```
$oneuser show
USER 0 INFORMATION
ID : 0
NAME : oneadmin
GROUP : oneadmin
PASSWORD : 3bc15c8aae3e4124dd409035f32ea2fd6835efc9
AUTH-DRIVER : core
ENABLED : Yes
USER TEMPLATE
TOKEN-PASSWORD="ec21d27e2fe4f9ed08a396cbd47b08b8e0a4ca3c"
RESOURCE USAGE  QUOTAS
```

Sunstone

Now you can try to log in into Sunstone web interface. To do this point login to your browser. If everything is OK you will be greeted with a login page. The user is oneadmin and the password is the one in the file /var/lib/one/.one/one-auth in your Front-end. If the page does not load, make sure you check /var/log/ one/ sunstone.log and /var/log/one/sunstone.error. Also, make sure TCP port 9869 is allowed through the firewall.

8.6.2 Node Installation

- **KVM Node Installation:**

  $sudo apt-get install opennebula-node

  $sudo service libvirtd restart # debian

  $sudo service libvirt-bin restart # ubuntu

- **Configure Passwordless SSH:**

  OpenNebula Front-end connects to the hypervisor Hosts using SSH. You must distribute the public key of oneadmin user from all machines to the file /var/lib/ one/.ssh/ authorized-keys in all the machines. There are many methods to achieve the distribution of the SSH keys, ultimately the administrator should choose a method (the recommendation is to use a configuration management

system). In this guide we are going to manually scp the SSH key

When the package was installed in the Front-end, an SSH key was generated and the authorized-keys populated. We will sync the id-rsa, id-rsa.pub and authorized-keys from the Front-end to the nodes. Additionally we need to create a known-hosts file and sync it as well to the nodes. To create the known-hosts file, we have to execute this command as user oneadmin in the Front-end with all the node names and the Front-end name as parameters:

$ssh-keyscan frontend node1 node2 node3 ... /var/lib/one/.ssh/known-hosts

Now we need to copy the directory /var/lib/one/.ssh to all the nodes. The easiest way is to set a temporary password to oneadmin in all the hosts and copy the directory from the Front-end:

- **Networking Configuration:**

  You may want to use the simplest network model that corresponds to the bridged drivers. For this driver, you will need to setup a linux bridge and include a physical device to the bridge. Later on ,when defining the network in OpenNebula

```
$scp -rp /var/lib/one/.ssh node1:/var/lib/one/
$scp -rp /var/lib/one/.ssh node2:/var/lib/one/
$scp -rp /var/lib/one/.ssh node3:/var/lib/one/
$...
```

  you will specify the name of this bridge and OpenNebula will know that it should connect the VM to this bridge, thus giving it connectivity with the physical network device connected to the bridge.For example, a typical host with two physical networks, one for public IP addresses (attached to an eth0 NIC for example) and the other for private virtual LANs (NIC eth1 for example) should have two bridges

```
$brctl show
bridge name bridge id STP enabled interfaces
br0 8000.001e682f02ac no eth0
br1 8000.001e682f02ad no eth1
```

- **Adding a Host to OpenNebula:**

  In this step we will register the node we have installed in the OpenNebula Front-end, so OpenNebula can launch VMs in it. This step can be done in the CLI or in Sunstone, the graphical user interface.

8.7 System Implementation

8.7.1 Client Side

The Client side module will mainly consist of the VM that is being deployed as a service from the Cloud Service Provider and a few client handler scripts which will be encompassed into an agent.

The agent is basically a collection of scripts and data handlers which will be used to efficiently facilitate data generation and transmission. As we are providing the service,the client/user will be able to customize the vm to his/her choice. This customization will happen on the Client module. The client will have certain restrictions as well as some rules that it must adhere to. All this would be incorporated into the Service Agreement which would be authenticated among both the parties prior to access of service.

The agent will be the sole component in generating and sending the logs to the server. These logs that our system will plan to handle are Network logs, Disk and Device logs and memory dumps. The agent will run the predecided commands and scripts continuously and keep updating these logs as and when required. The agent will also be involved with communication between the client and the server in order to transmit the logs at a regular interval for further segregation and analysis.

8.7.2 Server Side

There are three modules located at server side

1.Agent Manager

This module is responsible for communication with agents deployed at virtual machines,maintenance of unprocessed databases. This agent manager is completely written in java which that contains

1.A virtual Machine monitor:-Deals with Virtual machine.

2.Unprocessed Database

3.File Repository

2.Evidence Manager

This module is responsible for processing the unprocessed database and stores the results into processed database which is accessible to forensics experts. This module is written in java.

3.Access Portal

It is a web portal that serves as access point to processed database.It also maintains a details of who accessed a portal and when. Html and php(5.6) is used to develop this portal.

8.7.3 Databases

1.Unprocessed Database

Smart agent deployed at Virtual Machine traces the logs and other data that can be used as potential evidences in forensics mechanism.The Smart agent sends the data to server that has a huge database which stores the logs according to Vm deployed to various clients. This database contains various logs traced by agent such as process logs,File logs various communication and network logs,Port details and user details We have used mongodb (an unstructured database) due to varying nature of logs

2.Processed Database

Once a single iteration of data collection from virtual machine is done the data collected is processed in such way that unnecessary attributes of logs is removed.Some aggregation sorting and post processing is done. Such processed data which contains potential evidences are stored in separate database known as Processed Database. To store such processed data again we have used mongodb database.
The reason behind using mongodb especially unstructured database is varying nature of logs and huge amount of data collected from Virtual Machine.

3.File Repository

Apart from two databases described above a file repository is maintained.This repository holds all the files modified in Virtual Machine in a particular time limit.This repository will help forensics experts to get details of activities happened in Virtual Machine

8.8 Significance of Collected Logs

- **lsof -u $u_name :**

  lsof meaning List Open Files is used to find out which files are open by which process. As we all know Linux/Unix considers everything as a files (pipes, sockets, directories, devices etc.) One of the reason to use lsof command is when a disk cannot be unmounted as it says the files are being used. With the help of this command we can easily identify the files which are in use.

- **last -i –time-format iso :**

  Last searches back through the file /var/log/wtmp(or the file designated by the -f flag) and displays a list of all users logged in (and out) since that file was created. Names of users and ttys can be given, in which case last will show only those entries matching the arguments. Names of ttys can be abbreviated, thus

last 0 is the same as last tty0. When last catches a SIGINT signal (generated by the interrupt key, usually control-C) or a SIGQUIT signal (generated by the quit key, usually control-),last will show how far it has searched through the file; in the case of the SIGINT signal last will then terminate.

The pseudo user reboot logs in each time the system is rebooted. Thus last reboot will show a log of all reboots since the log file was created.

- **netstat -pntu :**

  netstat ("network statistics") is a command-line tool that displays network connections (both incoming and outgoing), routing tables, and a number of network interface (network interface controller or software-defined network interface) and network protocol statistics.

  It is available on Unix-like operating systems including OS X, Linux, Solaris, and BSD, and on Windows NT-based operating systems including Windows XP, Windows Vista, Windows 7 and Windows 8.

  It is used for finding problems in the network and to determine the amount of traffic on the network as a performance measurement.

- **ps aux :**

  The ps (i.e., process status) commandis used to provide information about the currently running processes, including their process identification numbers(PIDs).

  A process, also referred to as a task, is an executing (i.e., running) instance of a program. Every process is assigned a unique PID by the system.

  When ps is used without any options, it sends to standard output, which is the display monitor by default, four items of information for at least two processes currently on the system: the shell and ps. A shell is a program that provides the traditional, text-only user interface in Unix-like operating systems for issuing

commands and interacting with the system, and it is bash by default on Linux. ps itself is a process and it dies (i.e., is terminated) as soon as its output is displayed.

The four items are labeled PID, TTY, TIME and CMD. TIME is the amount of CPU (central processing unit) time in minutes and seconds that the process has been running. CMD is the name of the command that launched the process.

The -a option tells ps to list the processes of all users on the system rather than just those of the current user, with the exception of group leaders and processes not associated with a terminal. A group leader is the first member of a group of related processes.

The -u option tells ps to provide detailed information about each process. The -x option adds to the list processes that have no controlling terminal, such as daemons, which are programs that are launched during booting (i.e., computer startup) and run unobtrusively in the background until they are activated by a particular event or condition.

- **lsof -i :**

  Shows the list of all network connections LISTENING & ESTABLISHED.

- **find /home/{username} -mmin {time} -print :**

  The find command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions. By using the -exec other UNIX commands can be executed on files or folders found.

- **Stat -printf {required_format} :**

  The stat command pulls information from the file's inode. As you might be aware, there are actually three sets of dates and times that are stored for every

file on your system. These include the date the file was last modified (i.e., the date and time that you see when you use the ls -l command), the time the file was last changed (which includes renaming the file), and the time that file was last accessed.

- **tcpdump -n :**

  Tcpdump prints out a description of the contents of packets on a network interface that match the boolean expression; the description is preceded by a time stamp, printed, by default, as hours, minutes, seconds, and fractions of a second since midnight. It can also be run with the -w flag, which causes it to save the packet data to a file for later analysis, and/or with the -r flag, which causes it to read from a saved packet file rather than to read packets from a network interface. It can also be run with the -V flag, which causes it to read a list of saved packet files. In all cases, only packets that match expression will be processed by tcpdump.

  Tcpdump will, if not run with the -c flag, continue capturing packets until it is interrupted by a SIGINT signal (generated, for example, by typing your interrupt character, typically control-C) or a SIGTERM signal (typically generated with the kill(1) command); if run with the -c flag, it will capture packets until it is interrupted by a SIGINT or SIGTERM signal or the specified number of packets have been processed.

Chapter 9

SOFTWARE TESTING

9.1 Simulation of a Ping Attack

9.1.1 What Is A Ping Flood Attack

Ping flood, also known as ICMP flood, is a common Denial of Service (DoS) attack in which an attacker takes down a victim's computer by overwhelming it with ICMP echo requests, also known as pings.

The attack involves flooding the victim's network with request packets, knowing that the network will respond with an equal number of reply packets. Additional methods for bringing down a target with ICMP requests include the use of custom tools or code.

9.1.2 Attack Description

Normally, ping requests are used to test the connectivity of two computers by measuring the round-trip time from when an ICMP echo request is sent to when an ICMP echo reply is received. During an attack, however, they are used to overload a target network with data packets.

Executing a ping flood is dependent on attackers knowing the IP address of their target. Attacks can therefore be broken down into three categories, based on the target and how its IP address is resolved.

- **A targeted local disclosed ping flood** targets a single computer on a local network. An attacker needs to have physical access to the computer in order to discover its IP address. A successful attack would result in the target computer being taken down.

- **A router disclosed ping flood** targets routers in order to disrupt communications between computers on a network. It is reliant on the attacker knowing the internal IP address of a local router. A successful attack would result in all computers connected to the router being taken down.

- **A blind ping flood**

  involves using an external program to uncover the IP address of the target computer or router before executing an attack.

  There are a number of ping commands that can be used to facilitate an attack, including:

  - The n command, which is used to specify the number of times a request is sent.
  - The l command, which is used to specify the amount of data sent with each packet.
  - The t command, which is used to continue pinging until the host times out.

9.1.3 Attack Scenario

ICMP Ping Flood attack is a simple DoS attack where the attacker continuously sends a large amount of ICMP Echo Request (Ping) packets to the victim machine and saturates the network with traffic.

The response to each of these requests limits the amount of available system resources for other processes. The continuing requests and replies can be a reason for slowing the network and causing the legitimate traffic to continue at a significantly reduced speed or, in extreme cases, to be disconnected. A Ping Flood attack can effectively disable the network connectivity.

We have simulated a simple ping flood attack which done from a virtual machine to other pc. Agent runs every 2 minutes on the VM therefore it captures numerous ICMP packets using commands like tcpdump or netstat. These commands stores the data in the unprocessed database which is later processed by analyzer program along with other resources.

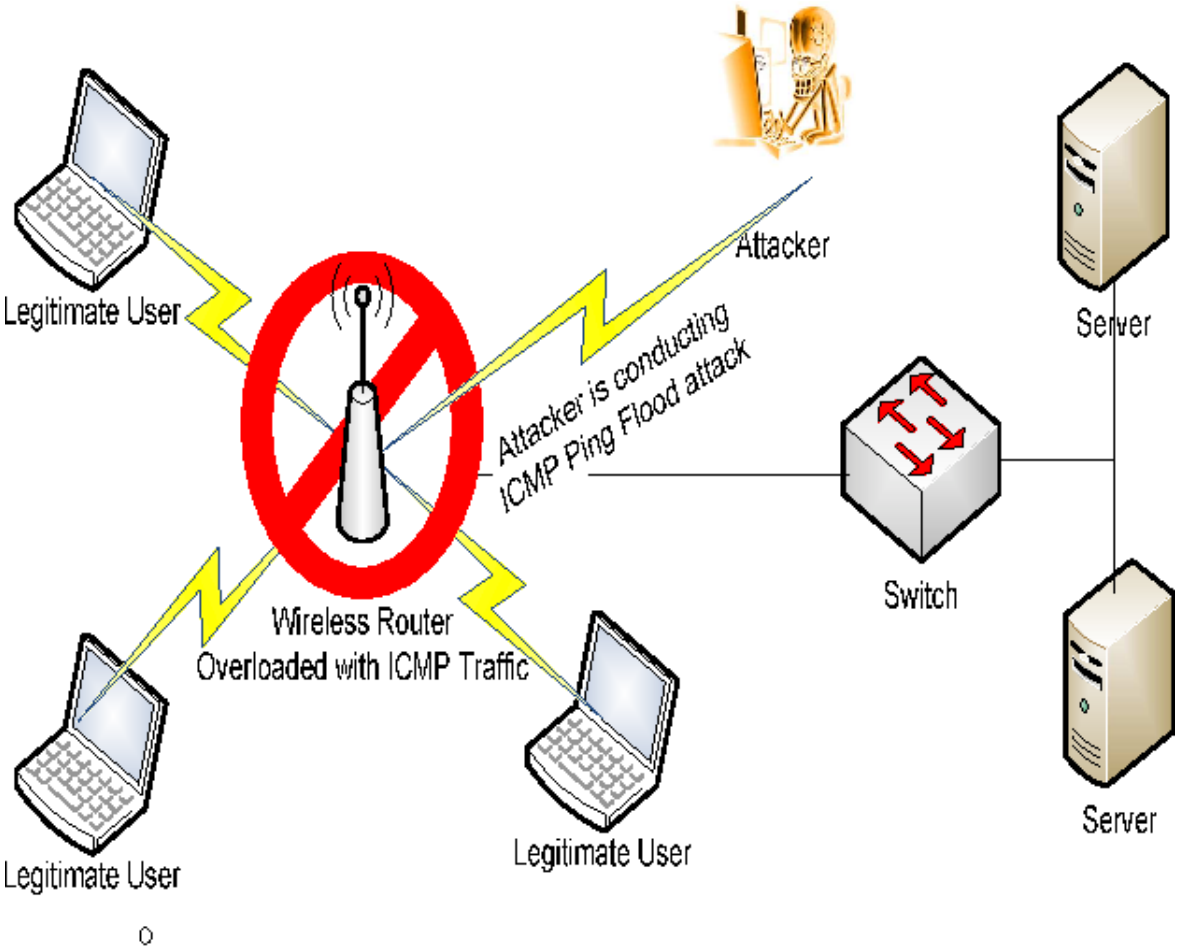


Figure 9.1: Attack Scenario

Given screenshot of our evidence portal shows the analyzed data of the time when the attack was done. It clearly shows the increased packet numbers. Thus our system can provide the way to prove the attack mishap. Forensic analyzer can use this data as a potential evidence.

9.2 Chain Of Custody

A critical part of any computer forensic investigation is ensuring proper evidence collection and proper maintenance of the chain of custody of the evidence collected. Positive control is the phrase most often used to describe the standard of care taken in the handling of potential evidentiary material (e.g., suspect computer systems, hard drives, and any backup copies). You need to be sure that you can identify the who, what, when, where, how, and why of each piece of evidence or material that you collect during the investigation

- **Who.**Who handled the evidence?
- **What.**What procedures were performed on the evidence?
- **When.**When was the evidence collected and/or transferred to another party?
- **Where.**Where was the evidence collected and stored?
- **How.**How was the evidence collected and stored?
- **Why.**For what purpose was the evidence collected?

Evidence Access Portal is collecting login details of each forensic expert who accesses the evidence portal

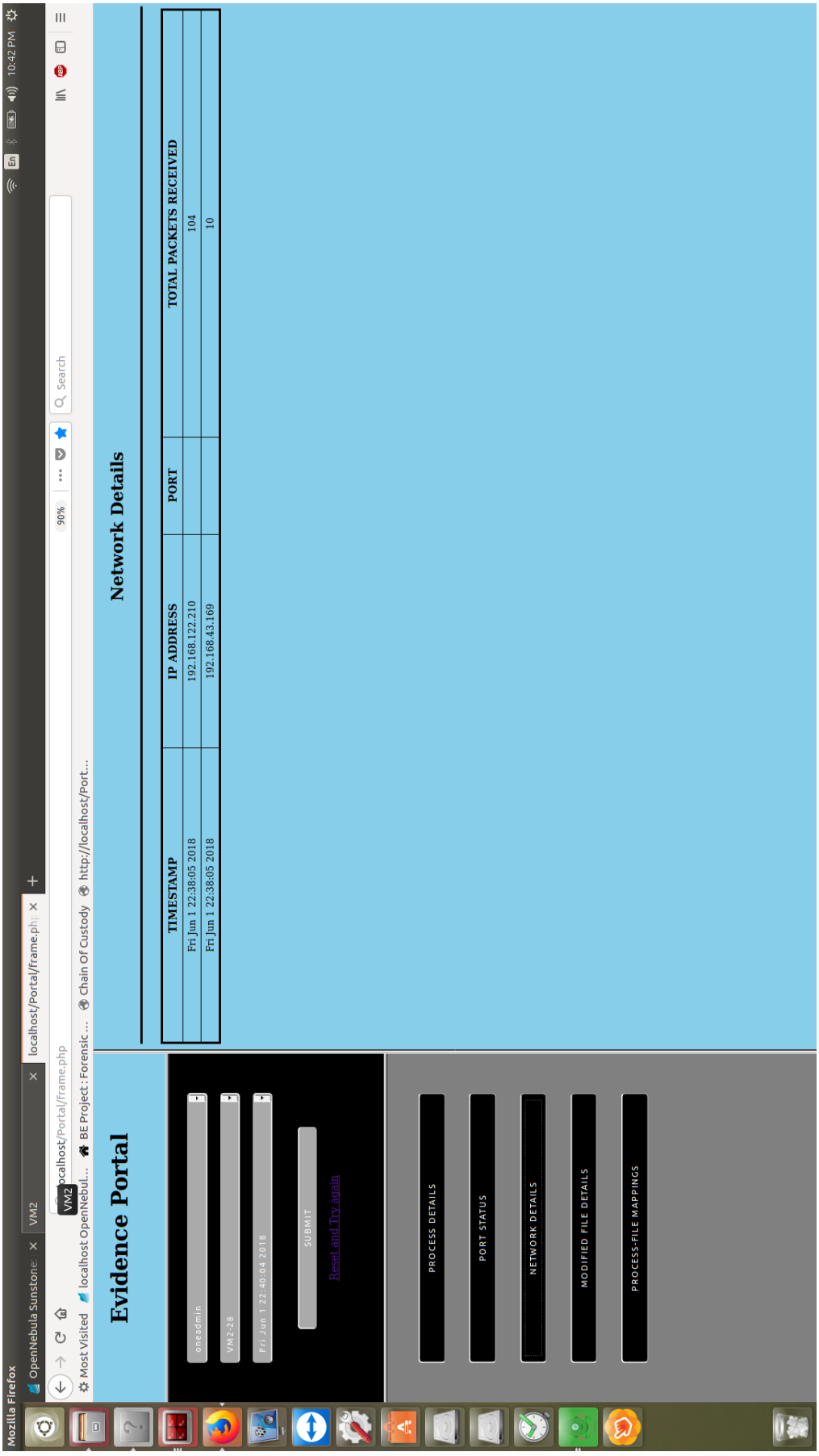


Figure 9.2: Analyzed logs showing increased packet count for a 2 min simulation

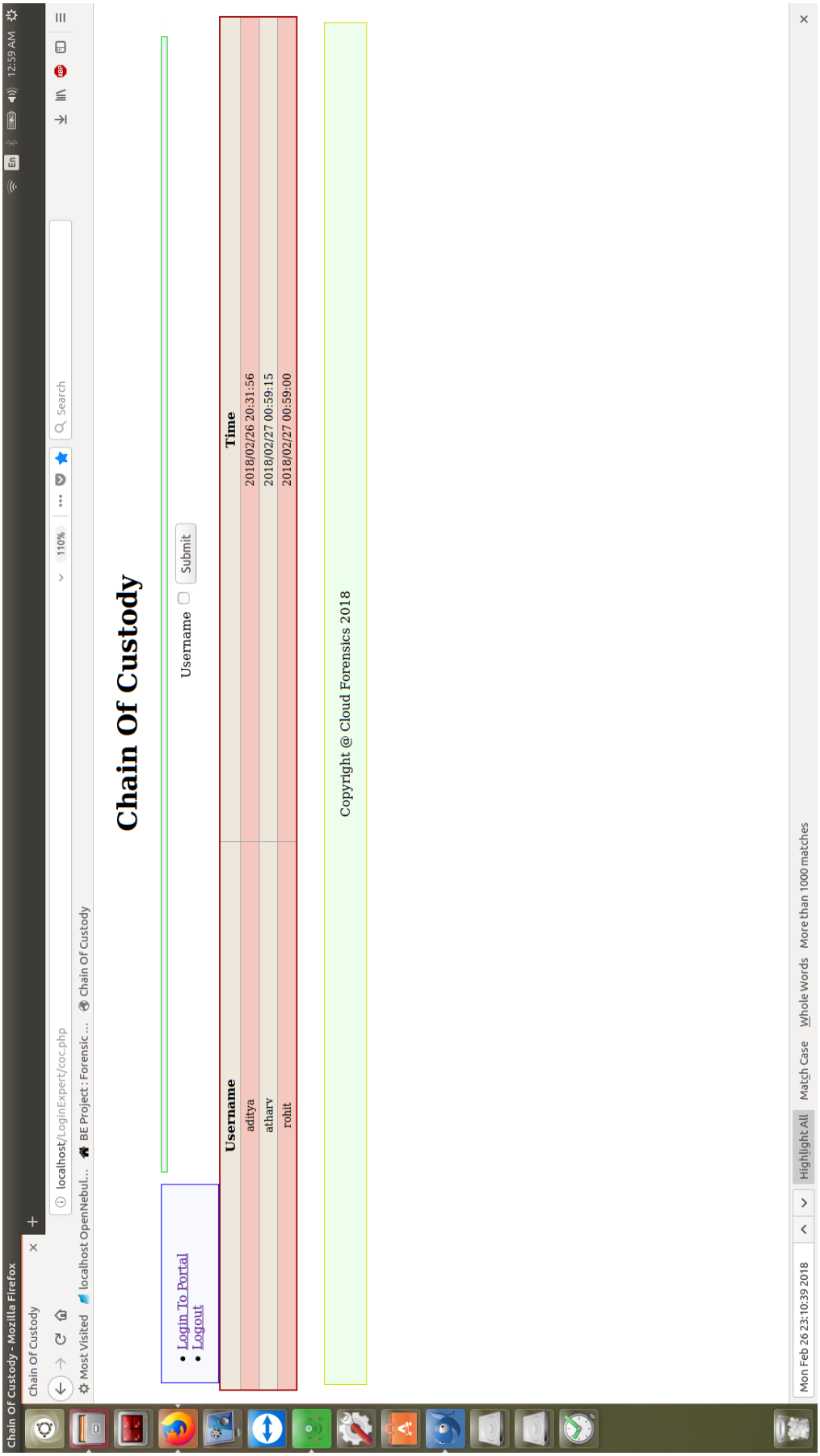


Figure 9.3: Chain Of Custody

Chapter 10

RESULTS

10.1 Screenshots

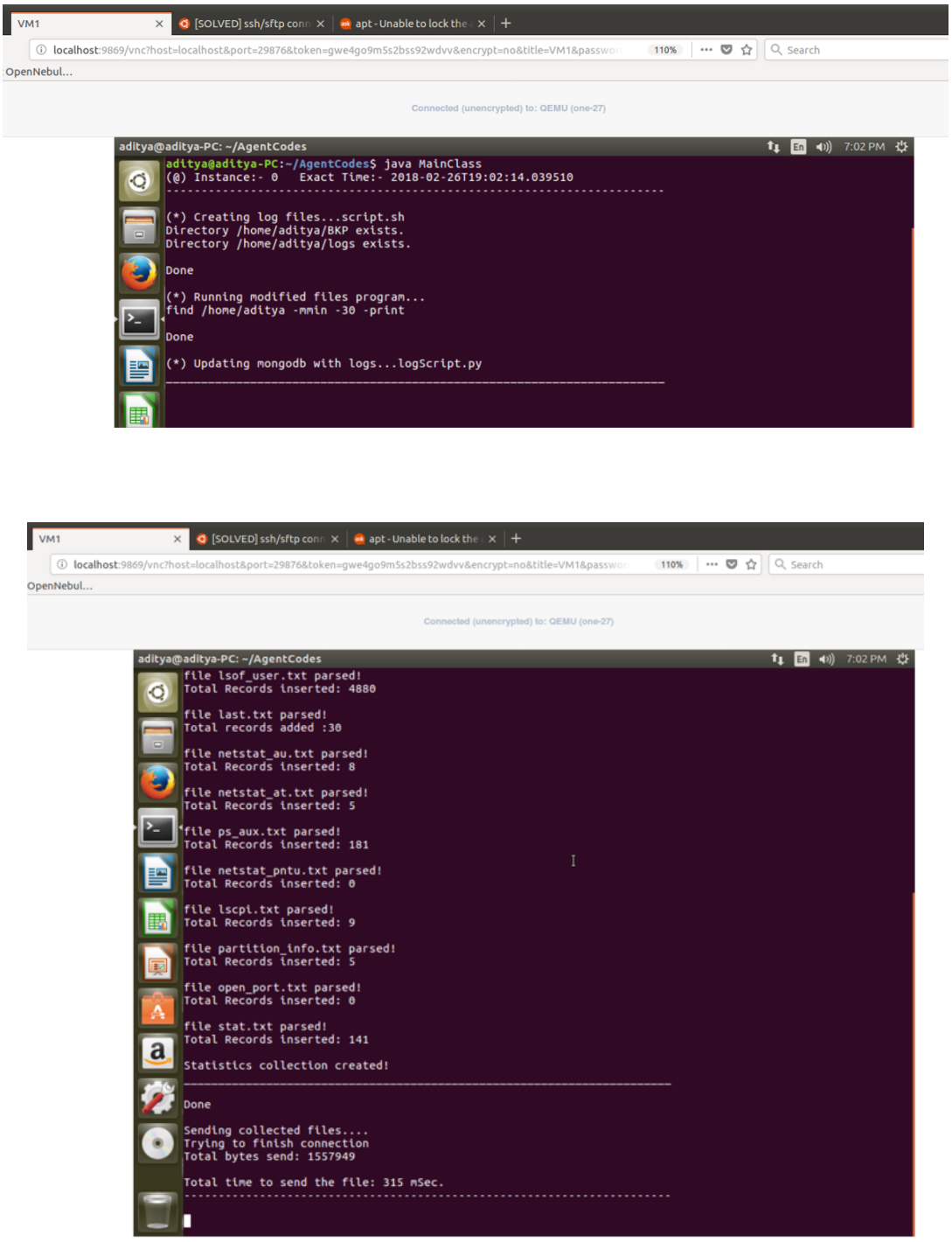


Figure 10.1: Smart Agent

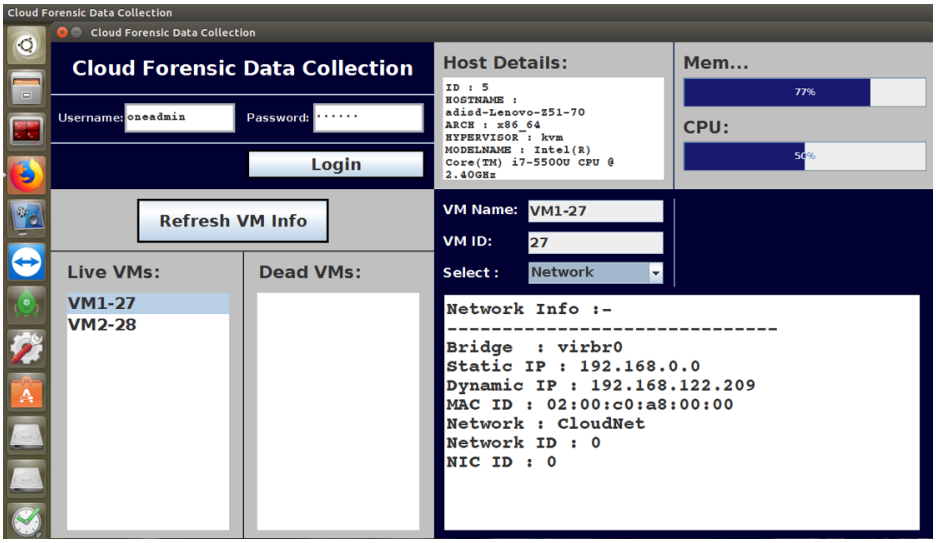
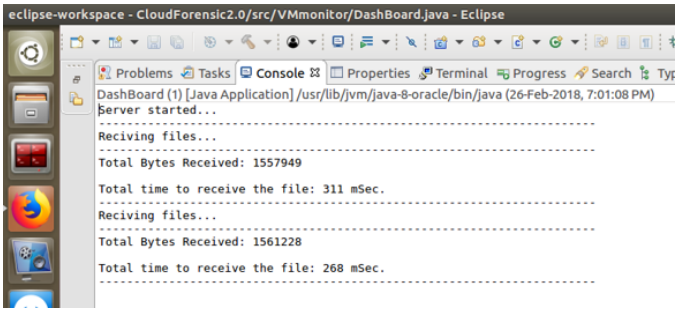
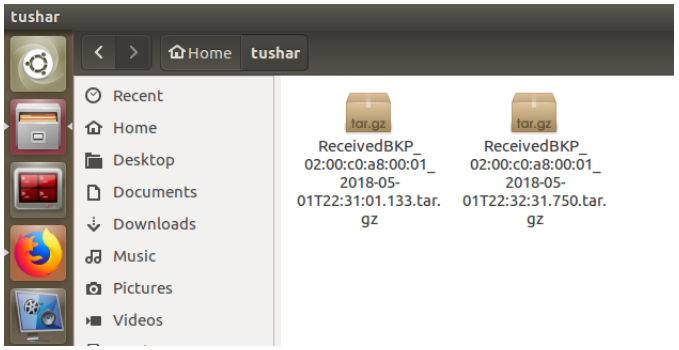


Figure 10.2: VM Monitor





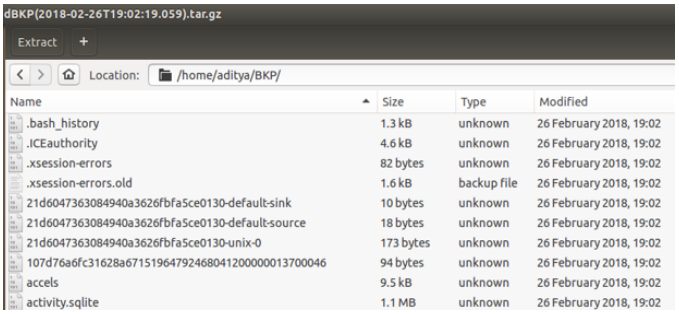


Figure 10.3: Components of VM Monitor

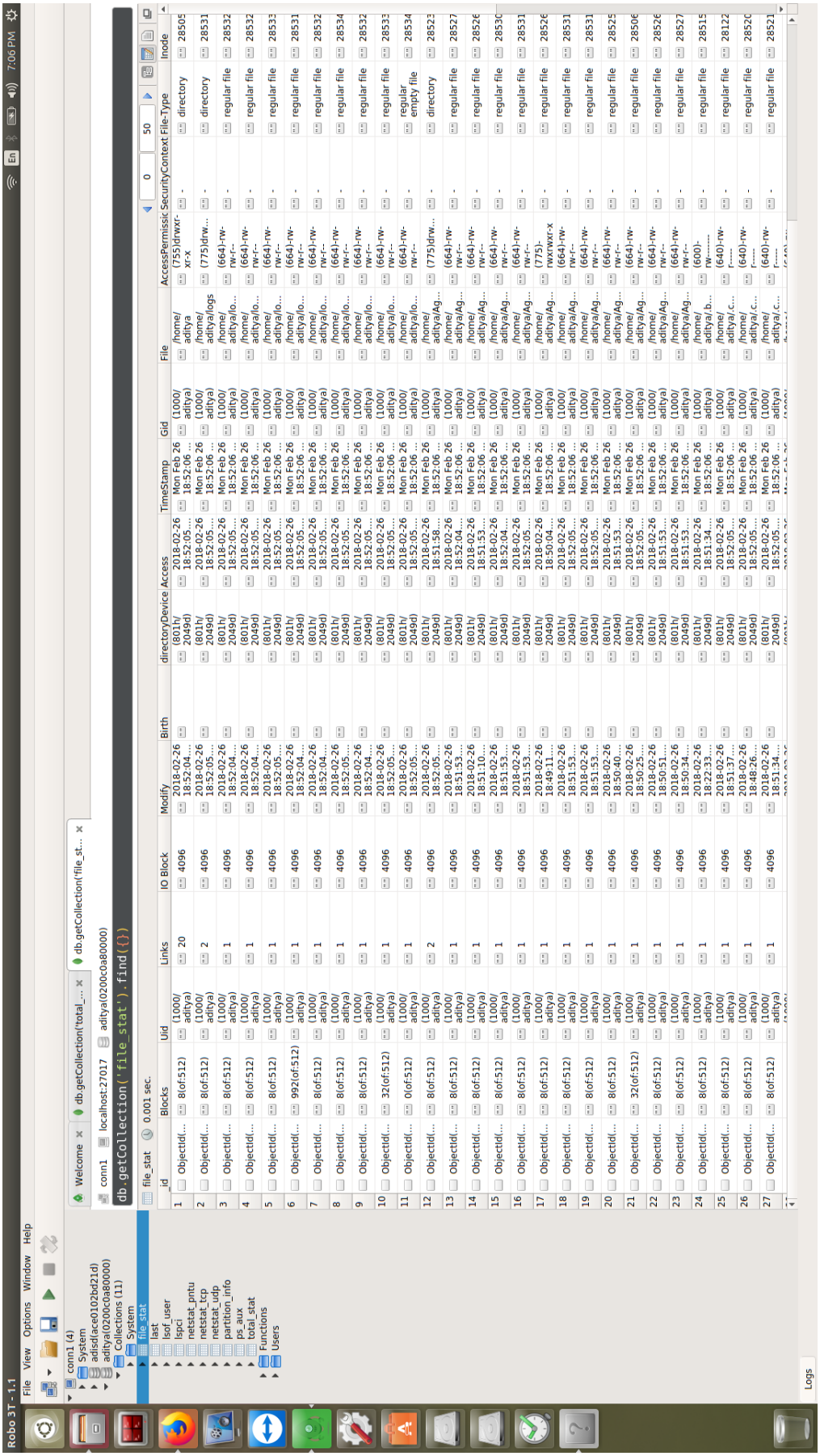


Figure 10.4: Unprocessed Database

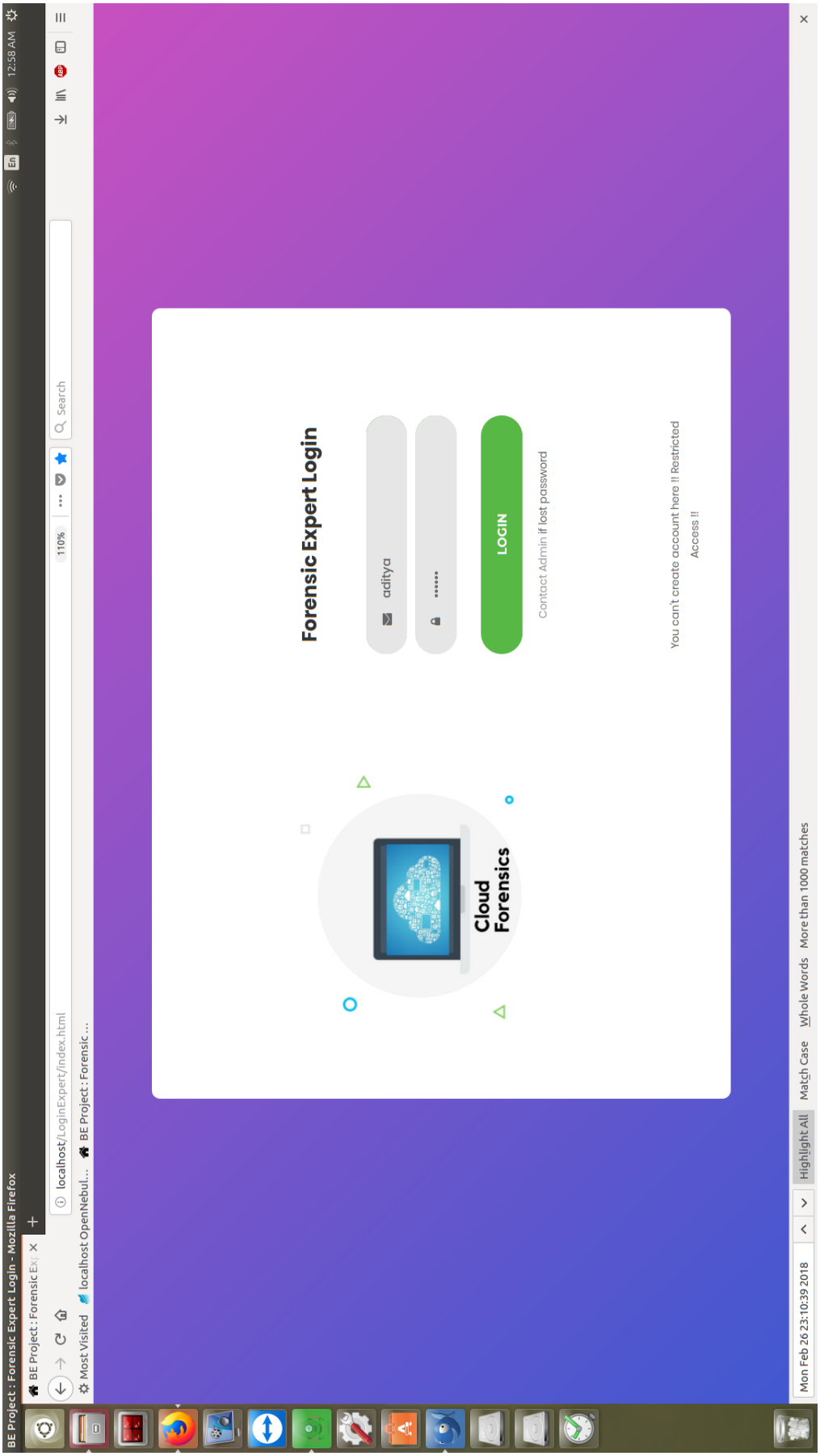


Figure 10.5: Evidence Access Portal Login

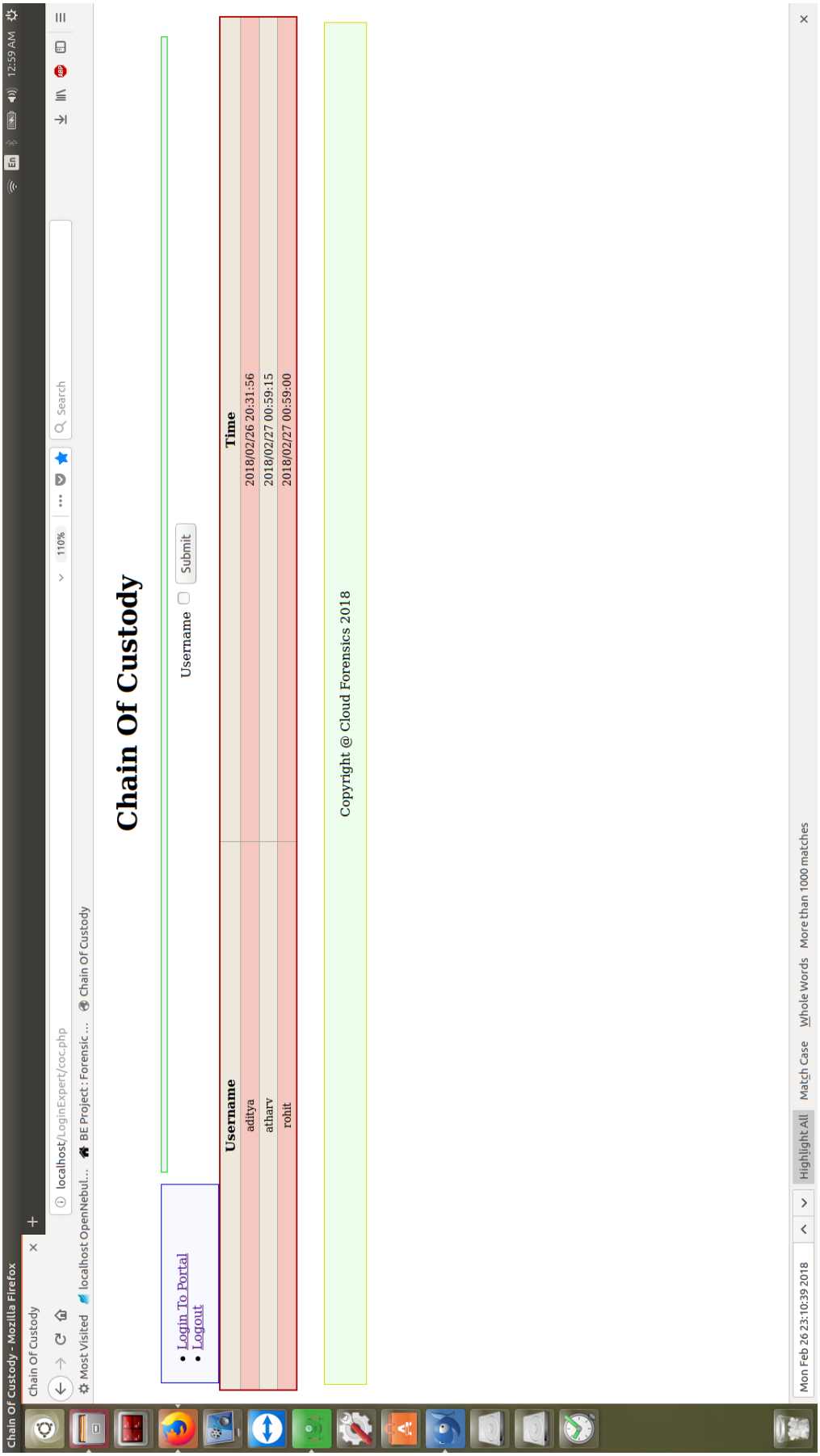


Figure 10.6: Chain Of Custody

Chapter 11

DEPLOYMENT AND MAINTENANCE

11.1 Installation And Un-Installation

11.1.1 Installation

- Server Side: OpenNebula 5.4 using commands
- Client Side: Users can purchase VMs from CSP. Authorized Forensic Experts will get login credentials.

11.1.2 Un-installation

- Server Side : Need to manually uninstall using commands.
- Client Side: User can request CSP to remove access and Forensic Experts can request to deauthorize their portal login details for revoking access.

11.2 User Help

Our web portal is quite simple.We have made it very user-friendly.The user can easily navigate through the pages of portal.

Chapter 12

FUTURE ENHANCEMENT AND CONCLUSION

12.1 Future Scope

In the future,we want to add the following as enhancement:

1. Applying Same Project to multi-cloud platform to solve forensics on it
2. Volatile memory analysis.
3. Incident Identification and Prevention.

12.2 Conclusion

In this project we are developing a standardize method for collecting and analyzing potential evidences . These evidences would help forensic experts to prove or disprove case .

Thus, we successfully solved the problem of collecting and analysing the potential evidences on private cloud.

In future, more functionalities can be added to use this application with multi cloud platform and valatile memory analysis.

Chapter 13

REFERENCES

Bibliography

[1] Cloud Forensics: A Meta- Study of Challenges, Ap- proaches, and Open Problems (Shams Zawoad , Ragib Hasan),2014

[2] RightScale 2017 State of Cloud Report (RightScale),2017

[3] Towards a security-enhanced PaaS platform for multi-cloud applications (Kyriakos Kritikos, Tom Kirkham, Bartosz Kryza, Philippe Massonet),2016

[4] NIST Cloud Computing Forensic Science Challenges (NIST),2014

[5] Towards Transparent and Trustworthy Cloud (Marco Anisetti, Claudio A. Ardagna, and Ernesto Damiani, DIUniversit degli Studi di Milano),2016

[6] Cloud-Trust - a Security As- sessment Model for Infras- tructure as a Service (IaaS) Clouds (Dan Gonzales, Member, IEEE, Jeremy Kaplan, Evan Saltzman, Zev Winkelman, Dulani Woods),2014

[7] Guide to Integrating Forensic Techniques into Incident Response (Karen Kent,Suzanne Chevalier, Tim Grance, Hung Dang),2006

[8] Cloud Log Forensics: Foun- dations, State of the Art and Future Directions (Suleman Khan, Abdullah Gani, Ain- uddin Wahid, Abdul Wa- hab, Mustapha Aminu Bagiwa, Muhammad Shiraz, Samee U. Khan, Ra- jkumar Buyya and Albert Y. Zomaya),2016

[9] Log Management in the Cloud: A Comparison of In- House versus Cloud-Based Management of Log Data (Jerry Shenk),2008

Annexure A

LABORATORY ASSIGNMENTS ON PROJECT ANALYSIS OF ALGORITHM DESIGN

- To develop the problem under consideration and justify feasibilty using concepts of knowledge canvas and IDEA Matrix.

  IDEA Matrix is represented in the following form. Knowledge canvas represents about identification of opportunity for product. Feasibility is represented w.r.t. business perspective.

I	D	E	A
Increase efficiency Of forensics evidence collection in Cloud	Drive Continuous monitoring of VM.	Evaluate potential evidence for forensic expert.	Accelerate forensics analysis on cloud platform
Improve evidence collection and analysis	Deliver secure cloud framework	Enhance identification of adversary through better forensics over cloud.	Associate traditional forensics tools and techniques
Ignore CSP trust issues	Decrease risk of losing potential evidence.	Eliminate Need of separate cloud forensics tools	Avoid Privacy violations of benign user

Table A.1: IDEA Matrix

- Project problem statement feasibility assessment using NP-Hard, NP- Complete or satisfy ability issues using modern algebra and/or relevant mathematical models.

  Polynomial(P) and Non-Polynomial(NP)

  - Polynomial Time:

    The class of polynomial- time solvable problems P includes all the sets in which membership may be decided by an algorithm whose running time is bounded by a polynomial.

    Example:

    * Linear Time-O(n)
    * Quadratic Time-O(n2)
    * Exponential Time-O(2n)
    * Logarithmic Time-O(logn)

  - Non-Polynomial Time:

    The class non-deterministic polynomially acceptable problems, NP contains all sets in which membership can be verified in polynomial time. All problems from P is also NP. Examples:

    * Decision problem version of the integer factorisation problem
    * Graph isomorphism problem
    * A decision version of travelling salesman problem
    * The Boolean satisfiability problem

    Types of NP type problems:

    * NP-hard:

      In computational complexity theory, is a class of problems that are,

informally, "at least as hard as the hardest problems in NP". More precisely, a problem H is NP-hard when every problem L in NP can be reduced in polynomial time to H.As a consequence, finding a polynomial algorithm to solve any NP-hard problem would give polynomial algorithms for all the problems in NP, which is unlikely as many of them are considered hard.

* NP-complete:

  In computational complexity theory, a decision problem is NP-complete when it is both in NP and NP-hard. The set of NP-complete problems is often denoted by NP-C or NPC. The abbreviation NP refers to "nondeterministic polynomial time".

Thus from the above definition our application of is P (Polynomial) class.

Annexure B

LABORATORY ASSIGNMENTS ON PROJECT QUALITY AND RELIABILITY TESTING OF PROJECT DESIGN

- Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify object, morphisms, overloading in functions (if any), and functional relations and any other dependencies (as per requirements). It can include Venn diagram, state diagram, function relations, i/o relations; use this to derive objects, morphism, overloading

With this strategy, a problem is solved by splitting it into sub-problems, solving them independently, and merging their solutions into a solution for the whole problem. The sub-problems can be solved directly, or they can in turn be solved using the same divide-and-conquer strategy, leading to an overall recursive program structure. The potential concurrency in this strategy is not hard to see: Since the sub-problems are solved independently, their solutions can be computed concurrently, leading to a parallel program that is very similar to its sequential counterpart.

Identify Morphism

Much of the terminology of morphisms, as well as the intuition underlying them, comes from concrete categories, where the objects are simply sets with some additional structure, and morphisms are structure-preserving functions.

A category C consists of two classes, one of objects and the other of morphisms. There are two objects that are associated to everymorphism, the source and the target.

For many common categories, objects are sets (usually with more structure) and morphisms are functions from an object to another object. Therefore the source and the target of a morphism are often called respectively domain and codomain.

A morphism f with source X and target Y is written f : X Y. Thus a

morphism is represented by an arrow from its source to its target. In the category of smooth manifolds, morphisms are smooth functions and isomorphisms are called diffeomorphisms.

Functional Dependency

Functional dependency is a relationship that exists when one attribute uniquely determines another attribute.

If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X-¿Y, which specifies Y is functionally dependent on X. Here X is a determinant set and Y is a dependent attribute. Each value of X is associated precisely with one Y value.

- Use of above to draw relevant Software modeling methods, techniques including UML diagrams or other necessities using appropriate tools.
- UML diagrams or other necessities using appropriate tools. UML diagrams consists of following diagrams:
    1. Use-Case Diagram
    2. Activity Diagram
    3. State Diagram
    4. Component Diagram
    5. Class Diagram
    6. Package Component Diagram
    7. Deployment Diagram
    8. Sequence Diagram

- Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability. Write also test cases [Black box testing] for each identified functions. You can use Mathematica or equivalent open source tool for generating test data.

Software testing can be stated as process of validating and verifying that a computer program/application/product:

- Meets the requirements that guided its design and development.
- Works as expected.
- Can be implemented with the same characteristics.
- Satisfies the needs of stake holders.

Software testing, depending on the testing methods employed, can be implemented at any time in the development process. Traditionally, most of the test efforts occur after the requirements have been defined and the coding process has been completed but in the agile approaches most of the test effort is ongoing. As such the methodology of test is governed by a chosen software development methodology.

Types of testing:

- **Unit testing:**

  Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming,

a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing.

– **Integration testing:**

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

– **Acceptance testing:**

Acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. It may involve chemical tests, physical tests, or performance tests. In systems engineering it may involve black-box testing performed on a system (for example: a piece of software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery.

Test Cases

1. Vm creation and some monitoring conditions
2. Availability of proper logs to generate evidence
3. Configuration of Settings.
4. Repository Analysis

GUI testing

GUI test is for testing if the GUI we have made for the application is working in the correct manner as desired. In our application we are going to have many buttons lists, tables so we will be checking if they are working properly.

Annexure C

PROJECT PLANNER

Table C.1: Plan of Project

From	To	Task	Status
27–06–2017	30–06–2017	Group Formation and finalization	Done
01–07–2017	15–07–2017	Topic Search	Done
16–07–2017	20–07–2017	Preliminary Information Gathering	Done
21–07–2017	28–07–2017	Project Dicsussion with Project Coordinator and topic finalization	Done
01–08–2017	04–08–2017	Synopsis preparation and submission	Done
25–08–2017 19–09–2017	30–08–2017 24–09–2017	Detailed Literature Survey .	Done .
25–09–2017	06–10–2017	Preparing Interim report	Done
20–12–2017	02–01–2018	Language Study	Done
03–01–2018	20–01–2018	Forensics Process analysis and Study	Done
21–01–2018	04–03–2018	Coding and Implementation	Done
05–03–2018	21–04–2018	Testing	Done
13–04–2018	21–04–2018	Final Documentation	Done
25–04–2018	20–05–2018	Final Project Report	Done

Annexure D

TERM-II PROJECT LABORATORY ASSIGNMENTS

1. Review of design and necessary corrective actions taking into consideration the feedback report of Term I assessment, and other competitions/conferences participated like IIT, Central Universities, University Conferences or equivalent centers of excellence etc.

   After presenting the design of our project, a few questions were asked regarding the detailing and feasibilty of its working. According to the feedback, given in the Term Assessment, the work on potential Evidence collections is completed. Also the representation of these evidence on web portal is done. Improved the design based on the techniques learnt in the subject Software Design Methodologies testing.

2. Project workstation selection, installations along with setup and installation report preparations.

   OpenNebula is used as the base for our project.

   OpenNebula provides the most simple but feature-rich and flexible solution for the comprehensive management of virtualized data centers to enable private, public and hybrid IaaS clouds. OpenNebula interoperability makes cloud an evolution by leveraging existing IT assets, protecting your investments, and avoiding vendor lock-in.

   OpenNebula is a turnkey enterprise-ready solution that includes all the features needed to provide an on-premises (private) cloud offering, and to offer public cloud services.

   Workstation: Lenovo i7 Series Laptops

   Operating System: Linux ubuntu 16.04

3. Programming of the project functions, interfaces and GUI (if any) as per 1 st Term term-work submission using corrective actions recommended in Term-I assessment of Term-work.

- Smart Agent:

  Smart agent collects the logs. It will monitor running processes and files updated by them. It will check recently updated files and send them to agent manager. Data will be encrypted before sending for security concerns. Encryption done by using SHA algorithm. Then encrypted data will be send to Agent Manager using SSH. Secure Shell(SSH) is acryptographic network protocol for operating network services securely over an unsecured network.The best-known example application is for remote log into computer systems by users. SSH provides a secure channel over an unsecured network in a client server architecture, connecting an SSH client application with an SSH server

  Programming Tools: Java,Python

  Shell Commands:lsof,netstat,find,ls,ps–aux

- Agent Manager:

  Agent manager communicates with Smart Agent via SSH. It is responsible for decryption of data received from Smart Agent. It creates evidence unprocessed database and stores data received from Smart Agent. It will monitor All VM and keeps track of Smart Agent Availability.

  Programming Tools: Java8

  Database: MongoDB 3.6

- Evidence Manager:

  Evidence Manager is responsible for analysis of unprocessed evidences.

Data will be formatted into relevant format and user wise organized. Finally, it will store potential evidences into evidence database

Programming Tools: Java8

Database: MongoDB 3.6

- Evidence Access Portal:

  Forensic experts will get access to potential evidences by using Evidence Access Portal. Experts authentication logs are maintained for chain-of-custody and data integrity

  Programming Tools: Java8,HTML,PHP5

  Database: MongoDB 3.6

4. Test tool selection and testing of various test cases for the project performed and generate various testing result charts, graphs etc. including reliability testing.

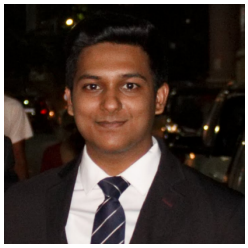
   Testing is a critical software development activity because it helps you improve the quality of your apps, ensure better user satisfaction, and reduce overall development time spent on fixing defects.

   (a) **Developer**

   A developer creates the bundle projects, defines the core definitions for them. Creates additional custom application code, builds the application, manual and automated tests of the application code by using the development tools. Optionally, the developer can use Java, JavaScript, Angular JS, HTML, or CSS to extend the power of the framework. Most of the core tasks of a developer can be carried out by using Innovation Studio, but extensions can be built using the IDE.

Annexure E

INFORMATION OF PROJECT GROUP MEMBERS



1. Name : Rohit Gund
2. Date of Birth :23/08/1996
3. Gender : Male
4. Permanent Address: Plot No. 15,Shahunagar, Rahuri-413705
5. E-Mail :rohitgund@gmail.com
6. Mobile/Contact No. :8888942400
7. Placement Details : Did not appear for placements
8. Paper Published : NO

1. Name : Atharv Vibhute
2. Date of Birth :08/04/1996
3. Gender : Male
4. Permanent Address:73,Middle Lane,Narsobawadi Dist-Kolhapur-416104
5. E-Mail : atharv.vibhute08@gmail.com
6. Mobile/Contact No. :7588489816
7. Placement Details :ZS Associates
8. Paper Published : NO

1. Name : Aditya Dhage
2. Date of Birth :31/10/1996
3. Gender : Male
4. Permanent Address:11,Sahajeevan Society, Ghule wadi road, Sangamner-422605
5. E-Mail : adhage93@gmail.com
6. Mobile/Contact No. :9850576949
7. Placement Details :Sigma OSS India Pvt Ltd
8. Paper Published : NO

1. Name : Tushar Chopade
2. Date of Birth :27/08/1996
3. Gender : Male
4. Permanent Address: Sr No.121,Lane no.3, Opp Aakash English Medium School,Khandave-Nagar,Pune-411014
5. E-Mail : tusharchopade27@gmail.com
6. Mobile/Contact No. : 8698857711
7. Placement Details :ATOS India
8. Paper Published : NO