Francois D'Ugard

vfrfrr



MISSION CRITICAL CLOUD COMPUTING

Requirements Document

# Copyright Notice

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation (THE SOFTWARE). THE SOFTWARE is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.  
  
THE SOFTWARE is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. In no event shall the authors be liable for any claim, damages, or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software. See the GNU General Public License for more details.  
  
You should have received a copy of the GNU General Public License along with this software. If not, see http://www.gnu.org/licenses.

This project will deliver mission assurance to mission-critical applications in cloud computing systems. We will do so by leveraging the unique capabilities that develop a virtual machine based approach to run applications with good security and reliability in typical cloud computing systems. This project will build upon the previous project's results:

1) A VM management system that dynamically migrate VMs across hosts on an Openstack-based cloud platform;

2) A P2P overlay network that interconnect the OpenStack VMs based on the IP-over-P2P (IPOP) framework. The project will focus on developing an extension to IPOP that will allow for the communications among the VMs to be routed by an overlay network in an OpenStack-based cloud system.

# Abstract

Virtualization is an increasingly popular approach to manage rising information technology costs and complexity in every sector of the economy. Cloud computing allows organizations of any size to provision infrastructure resources as needed and flexibly scale technology resources to meet changing demands. IaaS providers pool hardware resources and allocate them based on the requests of their users. In order to efficiently allocate these resources providers must aggregate users of different requirements and workloads onto the same physical infrastructure. However, this approach increases the likelihood that a malicious user can collocate a VM alongside a target VM in order to extract information or disrupt its functioning in some way.

We propose a solution that can deliver mission assurance to mission-critical applications in cloud computing systems. We will do so by leveraging the unique capabilities of virtualization technology that and develop a dynamic and distributed approach to run applications with good security and reliability in typical cloud computing systems.

Our approach relies on developing a moving-target defense (MTD) strategy, where virtual machines are randomly migrated across physical hosts within a cloud to dynamically determined locations. With the purpose of reducing the possibility of a malicious VM locating the mission-critical VM and achieving co-residency with it, or reduce the attacker’s co-residency time. In this document we evaluate the feasibility of several alternatives for the implementation of the proposed system and provide a high-level description of our plan

Table of Contents

[Copyright Notice 1](#_Toc397940240)

[Abstract 2](#_Toc397940241)

[1. Introduction 4](#_Toc397940242)

[1.1. Problem Definition 5](#_Toc397940243)

[1.2. Scope of System 6](#_Toc397940244)

[1.3 Overview of Document 7](#_Toc397940245)

[2. Current System 8](#_Toc397940246)

[3. Project Plan 9](#_Toc397940247)

[3.1. Project Organization 9](#_Toc397940248)

[3.2. Work breakdown 9](#_Toc397940249)

[3.3 Cost Estimate 11](#_Toc397940250)

[4. Proposed System Requirements 12](#_Toc397940251)

[4.1. High-level Definition of User Requirements 12](#_Toc397940252)

[4.2 Analysis of System Requirements 12](#_Toc397940253)

[4.2.1 Scenarios 12](#_Toc397940254)

[4.2.2 Use case model 12](#_Toc397940255)

[4.2.3 Static model e.g., object diagrams, class diagram 12](#_Toc397940256)

[4.2.4 Dynamic model e.g., sequence diagrams or state machines 12](#_Toc397940257)

[5. Glossary 13](#_Toc397940258)

[6. Appendix 14](#_Toc397940259)

[6.1 Appendix A - Complete use cases 14](#_Toc397940260)

[6.2 Appendix B - Use case diagram using UML 14](#_Toc397940261)

[6.3 Appendix C - Static UML diagram 14](#_Toc397940262)

[6.4 Appendix D - Dynamic UML diagrams 14](#_Toc397940263)

[6.5 Appendix E - User Interface designs. 14](#_Toc397940264)

[6.6 Appendix F - Diary of meeting and tasks. 14](#_Toc397940265)

[7. References 15](#_Toc397940266)

# Introduction

Virtualization is an increasingly popular approach to manage rising information technology costs and complexity in every sector of the economy. Cloud computing allows organizations of any size to provision infrastructure resources as needed and flexibly scale technology resources to meet changing demands. IaaS providers like Amazon, Microsoft, or Rackspace pool hardware resources such as compute, memory, and storage and allocate them based on the provisioning requests of their users and available resources. In order to efficiently and cost effectively allocate these resources providers must necessarily aggregate users of different requirements and workloads onto the same physical infrastructure.

The rising popularity and diffusion of hardware virtualization technology among organizations of every size has led researchers and technology professionals to seriously consider the security of such computer systems. By allocating the virtual machines of different user onto the same infrastructure various security vulnerabilities are created. However many of the unique capabilities of cloud computing can also be used to increase the reliability of the cloud as well as ensure its security and resilience to an attack.

An important quality of virtualization is the ability for users to quickly provision resources as needed. This capability lends itself well to the concept of migration wherein a virtual machine is suspended or shutdown and its resources are reallocated to another physical machine and then restarted or rebooted. This capability can be used to ensure the security of a cloud computing environment.

## Problem Definition

The rising popularity of cloud computing concepts has introduced significant security vulnerabilities into the information technology infrastructure of many organizations. Currently virtualized information technology systems are designed and built to operate on relatively static configurations. Typically cloud services providers fulfill provisioning requests on an as needed basis. In order for cloud computing to remain cost effective infrastructure providers must provision pools of resources, such as CPU, memory, and storage among many different users.

This configuration means that a malicious user can reside alongside a virtual machine that computes mission critical or sensitive data. The sharing of physical infrastructure among virtual machines belonging to different users opens up the possibility of a side-channel attack. A side channel attack occurs when a malicious users is able to locate a target virtual machine and spawn another virtual machine alongside this target. The malicious user is then able to extract information by monitoring the hardware resources shared by both virtual machines.

## Scope of System

Research has been conducted to explore the existing vulnerabilities in current third-party clouds by the sharing of the physical infrastructure among VMs belonging to different users, and analyze the practicality of cross-VMs side-channel attacks in such environments [7].

Moving Target Defense System research has proposed several MTD systems that change the structure of a cloud environment as well as its configuration. These systems operate autonomously and proactively to deny the opportunity for malicious users to exploit the side channel attack vulnerabilities inherent in public clouds.

We propose a proactive dynamic and random moving target defense system that leverages the capabilities of both live and non-live migration. Furthermore we propose a distributed peer to peer virtual private network that maintains the communication network structure of collaborating virtual machines unchanged as frequent migrations take place. Finally in order to increase the resiliency of said network we propose a secondary system the masks the communication between the virtual machines, making the communication patterns of virtual machines difficult to identify and further obfuscating the

## Overview of Document

This document is organized into various different sections. Chapter 1 is an introductory section. Section 1.1 defines and describes the problem we will solve. 1.2 provides some background information related to the problem including previous research. Section 1.3 we define and explain various terms and abbreviations found in this document.

Chapter 2 is a study into the feasibility of the proposed system. Section 2.1 describes the current system in place and identifies its current limitations and constraints. Section 2.2 presents the purpose of the system in detail. Section 2.3 is an overview of the user requirements that must be met upon the successful completion of the project. Based on those requirements section 2.4 compares other possible solutions and based on this analysis section 2.5 presents our recommended solution.

Chapter 3 is the project plan. The project plan describes the organization of the project in section 3.1 Also included in section 3.1.1 is the project personnel organization and hard ware and software resources in section 3.1.2. Lastly section 3.2 clearly states that tasks, milestones and deliverables expected upon completion of the project.

Chapter 4 is the appendix. It includes a project schedule in section 4.1. The feasibility matrix in section 4.2 a cost matrix section 4.3 and a diary of meetings in section 4.4

Finally Chapter 5 is a bibliographic reference of any works cited in this document.

# Current System

Currently virtualized information technology systems are designed and built to operate on static configurations. Typically cloud services providers fulfill provisioning requests as requested by their user base. In order for cloud computing to remain cost effective infrastructure providers must provision pools of resources, such as CPU, memory, and storage among many different users. This policy, while cost effective and efficient, opens up the cloud to the possibility of a side channel attack. By mapping the internal structure of the cloud environment a malicious attacker is able to identify a target virtual machine a spawn a malicious virtual machine co-resident to the target. This allows the malicious virtual machine to monitor the shared physical infrastructure and extract sensitive or mission critical data.

Existing Virtual Machine Monitors implement both live and no live migration techniques. However, these capabilities are usually intended to facilitate load balancing by moving workloads from one physical host to another to improve performance, or to serve as a maintenance tool, by relocating a hosts virtual machines in order to perform physical repairs. No current virtual machine monitors implement a live migration policy for the purpose of making the cloud more resilient or to improve the security of the cloud.

# Project Plan

This chapter presents the project organization including the personnel assigned to the project, and the mentor and client. We also specify the hardware and software resources required to complete the project. Lastly we identify the tasks, milestones and deliverables of our project and include a timeline of their completion dates.

## 3.1. Project Organization

This section describes the organization of all interested parties participating in the project. This section defines the roles and work domains of all interested parties in order to facilitate effective communication among the team members and the client. Also included in this section is a description of the hardware and software resources required for the successful completion of the project.

## 3.2. Work breakdown

Project Milestones

1. Complete setup of development environment and network configuration and setup NFS file system to support live migration capabilities of Openstack.
2. Migration Manager Component Design Complete
3. Communication network using IPOP P2P VPN Design complete
4. Implement Migration Manager Component
5. Implement Communications Network Component
6. Create Unit Tests for Migration Manager and Communications Systems
7. Integrate Communications Network Component into Openstack system

Deliverable Timeline:

9/8/2014

Feasibility Study and Project Plan. First draft.

Requirements Document. First draft.

Feasibility Study and Project Plan. In-class presentation.

9/15/2014

Feasibility Study and Project Plan. Second draft.

9/22/2014

Feasibility Study and Project Plan. Final draft.

Feasibility Study and Project Plan. In-class presentation

9/29/2014

Requirements Document. Second draft.

Requirements Analysis.

10/6/2014

Requirements Document. Final draft.

Requirements Analysis. In-class presentation.

Design Document. First draft.

10/20/2014

Design Document. Second draft.

System and Detailed Design. In-class presentation.

10/27/2014

Design Document. Final draft.

System and Detailed Design.

Implementation and Unit Testing. First code review.

11/17/2014

Implementation and Unit Testing. Second code review.

In-class presentation.

11/24/2014

Implementation and Unit Testing. Final code.

Implementation and Unit Testing.

Integration and System Testing. First code review.

## 3.3 Cost Estimate

|  |  |  |
| --- | --- | --- |
| **Resource** | **Quantity** | **Cost** |
| Server | 5 | $5,000 |
| Personal Computer | 3 | $2,400 |
| Software | Open Source | $0 |
| Personnel (Hours @ $40) | 480 | $19,200 |
| **TOTAL** |  | **$26,600** |

# Proposed System Requirements

## 4.1. High-level Definition of User Requirements

This section describes the requirements that the customer wants to include in the final software product and the constraints on its operation. The client cannot demand features outside the following list and the development team cannot claim completion of the project if it does not satisfy any of the items. The development team reserves the right to consider and negotiate additional features requested by the client.

* The system shall allow a user to dynamically migrate a VM across many hosts in an Openstack based cloud platform.
* The system shall allow the user to list all VM’s with the service enabled
* The system shall allow a user to enable the migration service for all vm’s
* The system shall allow a user to enable the migration service for a subset of vm’s
* The system shall allow a user to disable the migration service for all vm’s
* The system shall allow a user to disable the migration service for a subset of vm’s
* The system shall allow a user to create a P2P overlay network
* The system shall allow a user to destroy an overlay network
* The system shall allow a user to connect a vm to an existing P2P overlay network
* The system shall allow a user to disconnect a vm from a P2P overlay network
* The system shall expose both the migration and network services via a command line interface.
* The system shall allow a user to specify a configuration file that species all parameters for the overlay network.
* The system shall allow the user to specify a migration time range.
* The system shall schedule and execute migrations at random intervals based on the parameters specified by the user.
* The system shall allow a user to start, stop and restart the migration service at any time.
* The system shall support the execution of applications that use TCP and UDP protocols.
* The system shall prioritize the reconnection process with peers based on the traffic history.
* The system shall not constrain network infrastructure in order to function properly.

## 4.2 Analysis of System Requirements

Analysis models – contains the complete functional specification and is mainly for the designers and programmers. This section describes the diagrams in the Appendices B - D and validates the models against the use cases.

## 4.2.1 Scenarios

## 4.2.2 Use case model

## 4.2.3 Static model e.g., object diagrams, class diagram

## 4.2.4 Dynamic model e.g., sequence diagrams or state machines

# Glossary

API - Application Programming Interface: specifies how some software components should interact with each other [16].

Cloud computing: this phrase commonly refers to network-based services, which appear to be provided by real server hardware, and are in fact served up by virtual hardware that is simulated by software running on one or more physical machines [17].

Co-residency: a VM is co-resident with any VM when they are running on the same physical machine, and describes a great security risk when hosting VMs with sensitive data.

IP Address - Internet Protocol address: a numerical label assigned to each device participating in a computer network that uses the Internet Protocol for communication [19].

Hardware virtualization: this term refers to the creation of a virtual machine that acts like a real computer running an operating system [26].

LAN - Local Area Network: a computer network that interconnects computers in a limited area such as a home or a school using network media [21].

Live virtual machine migration: the process of moving a running virtual machine from a physical host to another physical host [6].

MTD - Moving-Target Defense: an approach that has been proposed to better protect important network systems and critical computing infrastructure by dynamically changing properties of their configuration in some way [1].

Non-live virtual machine migration: the process of moving a powered off virtual machine from a physical host to another physical host. After the transfer completes, the migrated virtual machine is restarted [6].

P2P network - Peer-to-peer network: a type of decentralized and distributed network architecture in which individual nodes in the network (called "peers") act as both suppliers and consumers of resources [22].

Private cloud: it is a cloud infrastructure operated solely for a single organization. It can be managed internally or by a third-party, and hosted internally or externally [17].

Public cloud: a cloud is said to be public when the services are rendered over a network that is open for public use, most commonly the Internet [17].

Side-channel attack: any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms [23].

UML - Unified Modeling language: a standardized, general-purpose modeling language in the field of software engineering. It includes a set of graphic notation techniques to create visual models of object-oriented software-intensive systems [25].

Virtual cluster: a group of VMs configured for a common purpose with associated storage resource, operating system, software environment, communication protocol, and network configuration [12].

VM - Virtual Machine: a software-based emulation of a physical computer [27].

VMM - Virtual Machine Monitor or hypervisor: a piece of computer software, firmware, or hardware that creates and runs virtual machines [20].

Virtual network: a computer network that consists of virtual network links as opposed to physical (wired or wireless) links between connected devices. It is implemented using methods of network virtualization [28].

VPN - Virtual Private Network: A VPN extends a private network across a public network, such as the Internet. It enables a computer to send and receive data across shared or public networks as if it were directly connected to the private network, while benefiting from the functionality, security and management policies of the private network [29].

# Appendix

## 6.1 Appendix A - Complete use cases

## 6.2 Appendix B - Use case diagram using UML

## 6.3 Appendix C - Static UML diagram

## 6.4 Appendix D - Dynamic UML diagrams

## 6.5 Appendix E - User Interface designs.

## 6.6 Appendix F - Diary of meeting and tasks.

# 7. References

[1] Carvalho, Marco, Jeffrey M. Bradshaw, et al. "Command and Control Requirements for Moving-Target Defense."IEEE. 1541-1672/12. (2012): 79-85. Print.

[2] Deshpande, Umesh, Xiaoshuang Wang, et al. "Live Gang Migration of Virtual Machines." International Symposium on High-Performance Parallel and Distributed Computing. (2011): 135-46. Print.

[3] Ganguly, Arijit, et al. “IP over P2P: Enabling Self-configuring Virtual IP Networks for Grid Computing,” Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International. IEEE, 2006.

[4] Haadi Jafarian, Jafar, Ehab Al-Shaer, and Qi Duan. "OpenFlow Random Host Mutation: Transparent Moving Target Defense using Software Defined Networking." ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. (2012): 127-32. Print.

[5] Ristenpart, Thomas, Eran Tromer, et al. "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds." ACM Conference on Computer and Communications Security. (2009): n. page. Print.

[6] Yackoski, Justin, and Scott A. DeLoach. "Mission-oriented Moving Target Defense on Cryptographically Strong Network Dynamics." Annual Cyber Security and Information Intelligence Research Workshop. (2011): n. page. Print.

[7] Zhuang, Rui, Su Zhang, et al. "Investigating the Application of Moving Target Defenses to Network Security." 6th International Symposium on Resilient Control Systems. (2013): 162-69. Print.

[8] Zhuang, Rui, Su Zhang, et al. "Simulation-based Approaches to Studying Effectiveness of Moving-Target Network Defense." National Symposium on Moving Target Research. (2012): n. page. Print.

[9] Wikipedia contributors. "Application programming interface." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 24 Jan. 2014. Web. 25 Jan. 2014.

[10] Wikipedia contributors. "Cloud computing." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 17 Jan. 2014. Web. 21 Jan. 2014.

[11] Wikipedia contributors. "IP address." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 21 Jan. 2014. Web. 22 Jan. 2014.

[12] Wikipedia contributors. "Hypervisor." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 7 Jan. 2014. Web. 21 Jan. 2014.

[13] Wikipedia contributors. "Local area network." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 16 Jan. 2014. Web. 21 Jan. 2014.

[14] Wikipedia contributors. "Peer-to-peer." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 13 Jan. 2014. Web. 21 Jan. 2014.