



Security Threats & Attacks on Virtualization for Cloud Computing

by
Kushan Sharma (108287X)

A thesis submitted to University of Moratuwa in partial fulfilment of the requirements for
the
Master of Computer Science, *Specialized in Computer & Network Security*

Department of Computer Science & Engineering
University of Moratuwa, Sri Lanka

December 2012



Security Threats & Attacks on Virtualization for Cloud Computing

by
Kushan Sharma (108287X)

A thesis submitted to University of Moratuwa in partial fulfilment of the requirements for
the
Master of Computer Science, *Specialized in Computer & Network Security*

Department of Computer Science & Engineering
University of Moratuwa, Sri Lanka

December 2012

Declaration

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any University or other institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Kushan Sharma:

Date

Approved by:

LtCol Dr Chandana D. Gamage
Department of Computer Science and Engineering
University of Moratuwa

Date

Copyright Statement

I hereby grant the University of Moratuwa the right to archive and to make available my thesis or dissertation in whole or part in the University Libraries in all forms of media, subject to the provisions of the current copyright act of Sri Lanka. I retrain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Kushan Sharma:

Date

I have supervised and accepted this thesis/dissertation for the award of the degree.

LtCol Dr Chandana D. Gamage
Department of Computer Science and Engineering
University of Moratuwa

Date

Abstract

Enterprises continuously seek innovative approaches to reduce operational computing costs while getting the most from their resources. Cloud Computing infrastructures are the latest technological advancement with the potential to maximize resource utilization while reducing costs.

The new paradigm of Cloud Computing possesses severe security risks to its adopters due to the distributed nature of Cloud Computing environments which make them a rich target for malicious individuals. Cloud infrastructure commonly relies on virtualization. The virtualization techniques used in Cloud possess numerous security threats and attacks. In order to cope with these risks, appropriate taxonomies and classification criteria for attacks on Cloud Computing are required. On the other hand Cloud Consumers runs numerous applications/scripts in order to complete their computing tasks. Most of them are too complex and complicated to trust. Even with access to the source code, it is difficult to reason about the security of these applications. They might harbor malicious code such as computer viruses, worms, bots, Trojan horses and spyware or contain bugs that are exploitable by carefully crafted input. It is essential that instead of just relying on conventional defense techniques, the next generation of system software must be designed from the ground-up to provide stronger isolation of services running on computer systems.

To address the above described security threats to Cloud Instances, we propose an architecture for confined execution environment to test untrusted applications/scripts inside Cloud Instances. Modern day security researchers consider malware sandbox analysis is as one of the promising approaches for exploring malware. But most of the previous proposed solutions have various security vulnerabilities due to the way of they have been implemented and the technologies that have been used in the implementation. The proposed architecture and proof-of-concept implementation address all the discovered drawbacks of previously presented sandbox solutions. We monitor all the system calls that are executed by the adversary to confine the adversary and limit the damage an attacker can cause to the Cloud Instance.

The research work related to the proposed sandbox architecture has been tested through LangshaJail, which is the proof-of-concept, built for the Cloud

Instances, using latest open source technologies that includes Linux as the Operating System Environment, Linux Resource Containers (LXC) as the virtualization environment and Seccomp as the system call filtering technology. The LangshaJail system was tested for compliance to Cloud Computing security objectives and adherence to performance criteria in order to validate the design approaches and implementation mechanisms used in the research.

Further as a part of the this thesis we present a taxonomy based on the notion of attack surfaces of virtualization for Infrastructure-as-a-Service-based Cloud offerings, thus making them more concrete and improving their analysis. The presented taxonomy specially addresses attacks based on residues of Cloud Instances. These residue based attacks are new venues for attackers that have previously not been addressed.

Acknowledgements

First of all I would like to thank my supervisor Dr. Chandana Gamage whose encouragement, guidance, support, and criticism from start to the very end, allowed me to understand the objectives and challenges of a master degree thesis. I would also like to thank Dr. Shehan Perera (Project Coordinator) for extensive advice, helpful feedback, and constant support.

Furthermore, my special thanks go to Prof. Gihan Dias, Dr. Shantha Fernando and Mr. Dileepa Lathsara from TechCERT who provided an excellent, supporting, innovative, and inspiring environment in which it was a pleasure to create this thesis. Last but not its a pleasure to thank all my TechCERT colleagues those who helped to make my thesis in a motion. I would also like to express my sincere gratitude to Mr Ben Grass from Vrije Universiteit Amsterdam for the technical expertise provided.

Finally, words alone cannot express the thanks I owe Mr. Dayananda Liyanage my father, Mrs. Kusuma Gamage my mother, Mr. Roshan Maduranga my brother, Mr. Hashan Dananjaya my brother and Ms. Umangee Chandrakumara my loving girl friend for all the encouragement extended.

Abbreviations

ABI - Application Binary Interface
ACL - Access Control List
API - Application Programming Interface
CC - Cloud Computing
CI - Cloud Instance
CISC - Complex Instruction Set Computing
COW - Copy-on-Write
CPU - Central Processing Unit
DLL - Dynamic Link Library
DNS - Domain Name System
FTP - File Transfer Protocol
GDT - Global Descriptor Table
HTTP - Hypertext Transfer Protocol
ICSPS - International Conference on Signal Processing Systems
IDT - Interrupt Descriptor Table
IDTR - Interrupt Descriptor Table Register
IPC - Inter Process Communication
IRC - Internet Relay Chat
ISA - Instruction Set Architecture
LDT - Local Descriptor Table
LXC - Linux Resource Containers
MMU - Memory Management Unit
MAC - Mandatory Access Control
MMU - Memory Management Unit
NIC - Network Interface Card
OS - Operating System
P2P - Peer-to-Peer
RISC - Reduced Instruction Set Computing
RM - Resource Monitor
SCSI - Small Computer System Interface
SIDT - Store Interrupt Descriptor Table
SDT - Software Dynamic Translation
SFI - Software-based Fault Isolation
SMTP - Simple Mail Transfer Protocol
SNMP - Simple Network Management Protocol

TCG - Trusted Computing Group
TCP - Trusted Computing Platform
TOCTOU - Time of Check to Time of Use
TPM - Trusted Platform Module
VM - Virtual Machine
VME - Virtual Machine Environment
VMM - Virtual Machine Monitor

Table of Contents

Declaration	v
Copyright Statement	vi
Abstract	vii
Acknowledgements	ix
Abbreviations	x
List of Tables	xvii
List of Figures	xviii
1 Introduction	1
1.1 Research Problem	2
1.2 Organization of the Thesis	3
2 Cloud Computing	5
2.1 Deployment Models of Cloud	6
2.1.1 Private Cloud	6
2.1.2 Public Cloud	7
2.1.3 Community Cloud	8
2.1.4 Hybrid Cloud	8
2.2 Delivery Models of Cloud	9
2.2.1 Infrastructure as a Service (IaaS)	9
2.2.2 Platform as a Service (PaaS)	10
2.2.3 Software as a Service (SaaS)	10
2.3 Characteristics of Delivery Models	11
2.3.1 On-demand Self-Service	11

2.3.2	Broad Network Access	12
2.3.3	Resource Pooling	12
2.3.4	Rapid Elasticity	13
2.3.5	Measured Service	13
2.4	Fundamental Security Challenges	14
2.4.1	Loss of Control Over Physical Assets	14
2.4.2	Incompatibility of Storage Devices	14
2.4.3	Secure Communication	14
2.4.4	User Developed Software	14
2.4.5	Patch Management	15
2.4.6	Security Compliance & Standards	15
2.4.7	Data Security	16
2.4.8	Virtualization Security	17
2.5	Reference Model for Cloud Computing	17
2.5.1	The Conceptual Reference Model	17
2.5.2	Scope of Control between Provider and Consumer	18
2.5.3	Research Interest within the Reference Model	19
2.6	Summary	20
3	Cloud Computing & Virtualization Security	22
3.1	Virtualization Security	22
3.2	A Taxonomy for Attacks on Virtualization for IaaS-based Cloud Computing	25
3.2.1	Flooding Attacks	25
3.2.2	Zombies in the Cloud	26
3.2.3	Cloud Malware Injections	28
3.2.4	Rootkit Inside VM Instances	29
3.2.5	Side Channel Attacks	31
3.2.6	Malicious Insiders	33
3.3	Proposed Solutions for Attacks on Virtualization for IaaS-based Cloud Computing	34
3.3.1	Virtualization & Hardware-based Security	34
3.3.2	Trusted Cloud Computing with Secure Resources and Data Coloring	35
3.3.3	Virtualization-Aware Malware Protection	36
3.3.4	Near Instant Recovery	36
3.4	Summary	37

4	Resource Isolation & Confined Execution	38
4.1	Taxonomy of Isolation Techniques	38
4.1.1	Language-based Isolation	39
4.1.2	VM-based Isolation	39
4.1.3	OS-kernel based Isolation	41
4.1.4	Hardware-based Isolation	41
4.1.5	Sandbox-based Isolation	41
4.1.6	VM-based vs Sandbox-based Isolation	42
4.2	Confined Execution with Sandboxes	43
4.2.1	Reference Model for Sandbox	43
4.2.1.1	Purely User-level Sandbox	43
4.2.1.2	Purely OS-based Sandbox	43
4.2.1.3	Hybrid Sandbox	44
4.2.2	A Survey of Sandboxing Techniques for Malware Analysis	46
4.2.2.1	Sandbox Analysis with Simulated Internet . .	46
4.2.2.2	Sandbox Analysis with API Hooking & DLL Injection	48
4.2.2.3	Sandbox Analysis with System Call Interposition	49
4.2.2.4	Sandbox Analysis with Isolated Network . . .	50
4.2.2.5	Sandbox Analysis with Standalone Computers	51
4.3	Summary	52
5	Virtualization & Sandbox Detection Techniques	54
5.1	Look for VME artifacts in processes, file system, and/or registry	56
5.2	Look for VME artifacts in memory	57
5.3	Look for VME-specific virtual hardware	57
5.4	Look for VME-specific processor instructions and capabilities .	58
5.5	Detection Countermeasures	58
5.6	Virtual Machine Residues	59
5.6.1	Carving Data of Virtual Machine Residues	59
5.6.2	Experiment Results	61
5.7	Summary	62
6	Building Sand Castles inside Cloud Instances	64
6.1	Motivation & Threat Model	64
6.2	Features for the Proposed Sandbox	67
6.3	Proposed Sandbox Architecture	70

6.4	Summary	73
7	Proof-of-Concept Implementation	74
7.1	Architecture of the Prototype	74
7.2	Execution of the Prisoner	77
7.2.1	Launching Program/Script to LangshaJail	77
7.2.2	Process Execution	78
7.2.3	Process Termination	78
7.3	Use of Existing Technologies	79
7.4	Building Sandbox Container	79
7.4.1	Namespaces	81
7.4.2	Control Groups	82
7.4.3	Controlled Access to Networking	83
7.4.4	Controlled Access to Memory	84
7.4.5	Controlled Access to Audio	85
7.4.6	Preserving Data Integrity using COW Storage	86
7.4.7	Life Cycle of LangshaJail	90
7.5	Filtering System Calls	90
7.5.1	Problematic System Call Groups	90
7.5.2	Avoiding Time of Check to Time of Use	92
7.5.3	Use of Seccomp	95
7.6	System Call Policy Generation	96
7.6.1	Initial Policy Template	97
7.6.2	Interactive Policy Generation	99
7.6.3	Structure of Policy	100
7.7	Summary	102
8	Conclusions	103
8.1	Security Performance Analysis	103
8.1.1	Detect Malware Using LangshaJail	103
8.1.2	Observation from Malware Behaviors	105
8.1.2.1	File Activity	105
8.1.2.2	Network Activity	106
8.1.2.3	Resource Monitoring	106
8.1.3	Thwarting the Detection of LangshaJail	106
8.2	Overall Review of Sand Castles for Cloud Instances	107
8.3	Future Work	108

List of Tables

2.1	Comparison between private Cloud and public Cloud	8
5.1	Summary of the result of each experiment	61

List of Figures

2.1	The Conceptual Reference Model [81, 49]	18
2.2	Scope of Controls between Provider and Consumer [81, 49] . .	19
4.1	Filtering architecture [40]	45
4.2	Delegating architecture [40]	46
4.3	Overview of the proposed system [143]	47
4.4	CWSandbox architecture overview [140]	49
4.5	Architecture of nicter [52]	51
4.6	Architecture of Cuckoo Sandbox [26]	52
5.1	Virtualized Server Environment used in experiment one	60
5.2	Virtualized Server Environment used in experiment two and three	60
6.1	Job hierarchy that contains a tree of processes labeled P0 through P7	68
6.2	Job hierarchy	68
6.3	Sandbox Reference Model	71
6.4	Syscall Decision Maker	72
7.1	Architecture inside the Jailer	75
7.2	Multiple instance of LangshaJail at a given time	75
7.3	General structure of a system call interception system	77
7.4	A prisoner's process hierarchy	79
7.5	Life Cycle of LangshaJail	90
7.6	Event Analysis of the vi Exploit [137]	93
7.7	Overview of system call interception and policy decision	101
8.1	A Screen Capture of LangshaJail Sandbox Container	104
8.2	A Screen Capture of Jailer Running Inside the Sandbox Container	104