
**UNIVERSITI PUTRA MALAYSIA**

# *DESIGN OF CLOUD-ENABLED CROSS-PLATFORM MALWARE ANALYSIS SYSTEMS*

**SEYED ABDOLRAHMAN MOUSAVIAN NAJAFABADI**

**FK 2017 131**

# DESIGN OF CLOUD-ENABLED CROSS-PLATFORM MALWARE ANALYSIS SYSTEMS

By

SEYED ABDOLRAHMAN MOUSAVIAN NAJAFABADI

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirements for the Degree of Master of Science

December 2016

**DEDICATIONS**

This thesis is dedicated to:
My parents, for their love, encouragement, and endless support;
And my friends who stood by me and supported me;
So this study has taken place today.

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Master of Science

**DESIGN OF CLOUD-ENABLED CROSS-PLATFORM MALWARE ANALYSIS SYSTEMS**

By

**SEYED ABDOLRAHMAN MOUSAVIAN NAJAFABADI**

**December 2016**

**Chair: Shaiful Jahari Bin Hashim, PhD**
**Faculty: Engineering**

The Internet of Thing (IoT) is already gaining momentum in the society by creating links between virtual technology and physical world. As the forecasts show, the number of devices connected to the Internet may rise to 100 billion devices by the end of the current decade. The dark side of this era, connecting everything to the Internet with lesser number security experts taking care of them. More importantly, companies are designing and implementing their platforms in the way that applications developed by third-party developers can be installed and executed seamlessly. It is to the best interest of the malicious attackers to violate the security and privacy by spreading malicious codes over a wider range of platforms including sensor nodes, smart phone, personal computer and server. This malicious activity utilizes zero-days vulnerabilities; thus the number of zero-days malware is expected to increase exponentially in the coming years. Arming security researchers with effective tools can lead to the discovery of malware in a shorter time. Hence we need an automated, cross-platform, scalable, fast, efficient and easy to use tools that can help even a novice user against the malicious attackers.

In this study, a demonstration of automated, cross-platform malware analysis system with the power of cloud computing in the form of Software-as-a-Service is proposed. An efficient technique is introduced to tweak the whole structure bottom up; from how the nodes should be arranged to create the network, to tune the performance of the computing resources (such as CPU, RAM, and hard disk), and to modifying all software running on top of this composition. The analysis engine is performed by an open-source dynamic malware analyzer called Cuckoo Sandbox which is not only modified and improved to perform

efficiently in the cloud environment but also able to support Android and Windows operating systems simultaneously. All the virtual machines that will be running the analysis are orchestrated by a fine-tuned OpenStack, an open-source cloud computing platform.

Results show that as the number of submitted jobs grow, the proposed and enhanced system works tremendously better than existing ones. By average, for Windows platform the measured consumed time to analyze and report the outcome is more than ten times faster than previous cloud-enabled malware analysis system and about twelve times faster than standalone version. For Android platform, on average the proposed system improved the performance four times faster than individual launch. Furthermore, the number of virtual machines that can be run in the whole system simultaneously has increased by seven times compared to the previous research system. The proposed and developed cross platform malware analysis system is operated autonomously with minimum intervention from the users.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia
sebagai memenuhi keperluan untuk ijazah Master Sains

**REKABENTUK SISTEM ANALISIS MAL-WARE DILENGKAPI CLOUD YANG MERENTAS-LANDASAN**

Oleh

**SEYED ABDOLRAHMAN MOUSAVIAN NAJAFABADI**

**Disember 2016**

**Pengerusi: Shaiful Jahari Bin Hashim, PhD**
**Fakulti: Kejuruteraan**

The Internet of Thing (IoT) sudah semakin mencipta nama dalam masyarakat hari ini dengan cara mencipta hubungan di antara teknologi maya dan dunia fizikal. Seperti yang diramal, bilangan alat yang disambung kepada Internet mungkin meningkat kepada 100 bilion di penghujung dekad ini. Dalam era yang gelap ini, semuanya dihubungkan kepada Internet dengan segelintir sahaja pakar keselamatan disediakan untuk menjaga dan melindungi pengguna. Apa yang lebih penting banyak syarikat sedang merekacipta dan melaksanakan landasan mereka dalam cara dimana aplikasi yang dibangunkan oleh pihak ketiga boleh dipasang dan digunakan tanpa sebarang masalah. Sudah menjadi hasrat penyerang untuk mengancam keselamatan dan privasi di Internet dengan menyebarkan kod-kod berbahaya merentas platform termasuk nod pengesan, telefon pintar, komputer peribadi dan pekhidmat (server). Aktiviti merbahaya ini menggunakan kelemahan hari-sifar; oleh itu bilangan malware hari-sifar dijangka akan meningkat dengan pesat dalam tahun-tahun akan datang. Menyediakan pengkaji-pengkaji keselamatan dengan alat efektif boleh membawa kepada penemuan malware dalam masa yang singkat. Maka, kita memerlukan satu set alat yang diotomatikan, rentas-landasan, boleh diskalakan, cepat, efisien dan mudah digunakan yang boleh membantu walau seorang pengguna baru sekalipun menentang penyerang-penyerang merbahaya ini.

Dalam kajian ini, satu demonstrasi sistem analisis yang diotomatikan, merentas landasan dengan kuasa perkomputeran cloud dalam bentuk perisian sebagai perkhidmatan telah dicadangkan. Satu teknik yang efisien yang telah diperkenalkan untuk memperbaiki keseluruhan struktur dari bawah; dari bagaimana nod perlu disusun-atur untuk mencipta jaringan, memperbaiki

prestasi sumber perkomputeran (seperti CPU, RAM, dan cakera keras), dan mengubahsuai semua perisian yang berfungsi di atas komposisi ini. Jentera analisis dijalankan melalui satu penganalisa malware dinamik sumber terbuka dipanggil Cuckoo Sandbox yang bukan sahaja diubahsuai dan diperbaiki untuk bekerja dengan lebih baik dalam persekitaran cloud tetapi ia juga mampu menyokong sistem Android dan Window serentak. Semua jentera maya yang akan menjalankan analisis tersebut dirangka oleh satu OpenStack, satu landasan perkomputeran cloud bersumber-terbuka yang telah disesuaikan.

Keputusan menunjukkan bahawa apabila bilangan tugas bertambah, sistem yang disarankan dan diperbaiki bekerja dengan lebih baik dari sistem sedia ada. Secara purata, untuk landasan Windows, masa yang digunakan dan disukat untuk menganalisa dan melaporkan hasilnya adalah sepuluh kali lebih pantas dari sistem analisis malware terdahulu dan dua belas kali lebih pantas dari versi yang boleh berdiri dengan sendiri. Untuk landasan Android, secara purata, sistem yang disarankan memperbaiki lagi prestasi empat kali ganda lebih pantas dari pelancaran individu. Tambahan lagi, bilangan mesin maya yang boleh dijalankan dalam kesemua sistem telah meningkat tujuh kali berbanding dengan sistem kajian terdahulu. Sistem analisis yang disarankan dan dibangunkan dioperasikan secara autonomi dengan campur tangan yang minima dari pengguna.

**ACKNOWLEDGEMENTS**

First and foremost, I would like to thank Almighty Allah (S.W.T) for giving me the strength, patience, courage, and determination to complete this work. All grace and thanks belong to Almighty Allah (S.W.T).

Many special thanks go to my supervisor Dr. Shaiful Jahari bin Hashim, for his incredible guidance, continuous support, and encouragement. He always had time for me and readily providing his technical expertise throughout the period of my study. I owe more than I can ever repay. The completion of this work becomes possible due to his supervision. His high stance of diplomatic power and professionalism set a great model for me to follow.

I would also like to thank my supervisor committee for serving on my thesis committee. Their helpful suggestions and advice on various aspects of my research work have certainly been very constructive. Without their kind cooperation and support, my graduate study would not have been accomplished.

I would also like to include an acknowledgment to my colleague, Hamidreza Hasheminejad. He provided me valuable advice and positive critics during my candidature. He guided me during the implementation of my work.

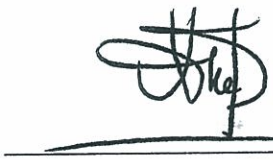
I certify that a Thesis Examination Committee has met on 19 December 2016 to conduct the final examination of Seyed Abdolrahman Mousavian Najafabadi on his thesis entitled "Design of Cloud-Enabled Cross-Platform Malware Analysis Systems" in accordance with the Universities and University Colleges Act 1971 and the Constitution of the Universiti Putra Malaysia [P.U.(A) 106] 15 March 1998. The Committee recommends that the student be awarded the Master of Science.

Members of the Thesis Examination Committee were as follows:

**Ahmad Shukri bin Muhammad Noor, PhD**
Associate Professor
Faculty of Engineering
Universiti Putra Malaysia
(Chairman)

**Fazirulhisyam bin Hashim, PhD**
Senior Lecturer
Faculty of Engineering
Universiti Putra Malaysia
(Internal Examiner)

**Massudi Mahmuddin, PhD**
Senior Lecturer
Universiti Utara Malaysia
Malaysia
(External Examiner)



**NOR AINI AB. SHUKOR, PhD**
Professor and Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 22 March 2017

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Master of Science. The members of the Supervisory Committee were as follows:

**Shaiful Jahari b. Hashim, PhD**
Associate Professor
Faculty of Engineering
Universiti Putra Malaysia
(Chairman)

**Abdul Rahman b. Ramli, PhD**
Associate Professor
Faculty of Engineering
Universiti Putra Malaysia
(Member)

**Khairulmizam b. Samsudin, PhD**
Senior Lecturer
Faculty of Engineering
Universiti Putra Malaysia
(Member)

**ROBIAH BINTI YUNUS, PhD**
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:

**Declaration by graduate student**

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any other institutions;
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and Innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software.

Signature: _____ Date: _____

Name and Matric No.: Seyed AbdolRahman Mousavian Najafabadi, GS37207

**Declaration by Members of Supervisory Committee**

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) are adhered to.

Signature: _____
Name of
Chairman of
Supervisory Committee: _____

Signature: _____
Name of
Member of Supervisory
Committee: _____

Signature: _____
Name of
Member of Supervisory
Committee: _____

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

| | |
|---|---|
| API | Application Programming Interface |
| APK | Android Application Package |
| ARM | Advanced RISC Machines |
| ATA | Advanced Technology Attachment |
| AV | Anti-Virus |
| AWS | Amazon Web Services |
| BSON | Binary Structured Object Notation |
| C&C | Command and Control |
| CD | Compact Disc |
| CERN | Conseil Européen pour la Recherche Nucléaire |
| CFG | Control Flow Graph |
| DHCP | Dynamic Host Configuration Protocol |
| DLL | Dynamic-link Library |
| EC2 | Elastic Compute Cloud |
| ESX | Elastic Sky X |
| ESXI | Elastic Sky X Integrated |
| EXT3 | Third Extended File system |
| EXT4 | Fourth Extended File system |
| GNU | GNU's Not Unix! |
| GPL | General Public License |
| GPS | Global Positioning System |
| HAAS | Honeynet as a Service |
| HDD | Hard Disk Drive |
| HTML | Hyper Text Markup Language |
| HTTP | Hyper-Text Transfer Protocol |
| HVM | Hardware Virtual Machine |
| I/O | Input/Output |
| IAAS | Infrastructure as a Service |
| IBM | International Business Machines |
| IDE | Integrated Drive Electronics |
| IDS | Intrusion Detection Software |
| IOT | Internet of Things |
| ISO | International Organization for Standardization |
| IT | Information Technology |
| JVM | Java Virtual Machine |
| KVM | Kernel Virtual Machine |
| LDAP | Lightweight Directory Access Protocol |
| LVM | Logical Volume Manager |
| LXC | Linux Containers |
| MB | Megabyte |
| MD5 | Message Digest 5 |
| NASA | National Aeronautics and Space Administration |
| NAT | Network Address Translation |
| NIC | Network Interface Card |
| NIST | The National Institute of Standards and Technology |
| PAAS | Platform as a Service |
| PCAP | Packet Capture |
| PIL | Python Imaging Library |

| | |
|---|---|
| QCOW2 | Qemu Copy On Write Version 2 |
| QEMU | Quick Emulator |
| RAID | Redundant Array of Independent Disks |
| RAM | Random Access Memory |
| REST | Representational State Transfer |
| RJE | Remote Job Entry |
| RPC | Remote Procedure Call |
| SAAS | Platform as a Service |
| SATA | Serial Advanced Technology Attachment |
| SCSI | Small Computer System Interface |
| SDK | Software Development Kit |
| SDN | Software-Defined Networking |
| SHA1 | Sha-1 Hash File |
| SMS | Short Message Service |
| TB | Terabyte |
| VM | Virtual Machine |
| VMM | Virtual Member Manager |
| VPN | Virtual Private Network |
| XML | Extensible Markup Language |
| ZFS | Zettabyte File System |

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

In the early 1970's the ARPANET was funded by Defense Advanced Research Projects Agency (DARPA) to meet military needs before it being expanded in public. Initially it was a private network services accessible only by few operational organizations, trying to connect limited number of computers together via Host-to-Host protocol. In that time, there was no actual threat because of the fact of being private and having limited number of computers. Paul Innella in a report is said that the first sign of a network security threat was seen in autumn 1988. It was a virus that infected about 60,000 computers connected to the Internet and cause a failure for at least two days.

Since then, every day huge amount of users and devices from a different type of categories, from normal clients to professionals and from private organizations to governments, are threatened by various viruses, worms, ransomware, and Trojans. Malware and viruses play a significant role in existing threats for computer users such as money loss, theft of personal data, accessing to the hardware without permission, etc. For example, Stuxnet which aimed to damage uranium enrichment infrastructure in Iran, or a Trojan-style malware that stole Angela Merkel's emails are two major malware that affect governments. Another indication of the problem is that even people without any special interest in computers are aware of worms like CryptoLocker or Sasser. To confront these incidents, experts in security field are required to create, develop, and use tools so that they can prevent data leak and loss, defend against attacks, discover vulnerabilities, and patch bugs quickly.

The appliances that researchers use to explore the behavior of malware are mainly divided into two groups: static analysis tools and dynamic ones. Malware Sandboxing is a pragmatic software that represents dynamic analysis method which instead of statically analyzing the source code and the binary files, it executes the files and monitors their behavior in real-time. The process of preparing a proper environment, running the binary file in that isolated surroundings, observing and collecting data and cleaning up the host can be automated to speed up analyzing flow. This study presents an enhanced approach to boost this flow by utilizing cloud computing power.

This chapter is organized as follows. Section 1.2 describes the motivation behind this study and the existing research problems. In Section 1.3 the aims and objectives of this study are stated. Section 1.4 outlines the scopes of the thesis followed by the thesis organization and reminder of coming chapters that is described in Section1.5.

## 1.2 Motivation and Problem Statement

Internet security teams at Symantec reported [1] that in 2014 more than three hundred million new malware were created which means almost one million new threats were distributed per day. In another report [2], they have stated that the number of zero-day vulnerabilities discovered in 2015 has increased by 125% compared to the previous year. This figure for Ransomware is 35 percent, for the personal records (including medical records) that have been stolen is 23% and 85% (half a billion personal records) for unreported exposures [2]. Facing this huge number of malware while most of the electronic devices in our homes are connected to the Internet leads us to this conclusion that researchers and defenses should move as fast as attackers. Analyzing new files quicker and more accurate than before can be a big step regarding this.

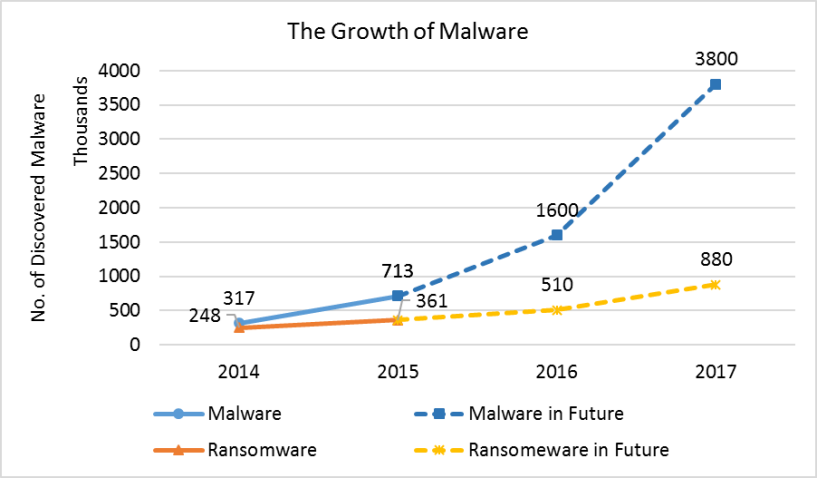



**Figure 1.1: The Growth of Malware**

Talking about the gradually growing number of smartphone users in last decade is stating the obvious but should never be forgotten as attackers never forgot this fact. In 2015, more than 50 percent of the Internet users were using Mobiles other than Desktops or other devices [3]. On the other hand, users are able to install applications on their device not only from Google Play Store but also from third-party markets or other websites known as “side-loading.” Side-loading is not avoidable for some users due to some restrictions of Google Play Store such as legal restrictions, region availability, and forcing developers to register with Google, etc. Hence, the potential of having vulnerabilities and loophole because of lack of verification is so high. Thus focusing on the files and malware that targeted smartphones are just not less important as other files.

Hardware is an essential part of any system in order to have enough resources to analyzing files. Scaling up the infrastructure by adding extra resources such as Servers, RAMs, CPU, etc. is not only pricey but also can save the time of non-blocking processes which is as efficient as one tiny floodgate for a huge dam. On the contrary, scaling the server's architecture out using cloud computing technology can give the advantage of sharing tasks and processes between different hypervisors.

The other subject that has influenced the design of this automated and scalable analysis approach is resetting the analysis environment to a clean state which is undoubtedly a time-consuming process. This refreshing process also is a necessity for executing every sample in an identical environment to consequently having a comparable and valid results. Deducting every seconds of this resetting process, can have a huge impact of the whole analysis process as the number of samples grow [4].

The manpower required to maintain the system or extending the platform in the time of need is expensive in the result of spending money on salaries. Moreover, existing automated malware collection systems like Honeypots can collect a tremendous number of malicious files from the Internet. The process of analyzing these data can be fully automated and also completely integrated with other systems by the power of cloud platform in the favor of processing in real time.

While existing solutions, such as [5], [6], [7] and [8], in the subject of analyzing huge amount of malwares have tried to resolve this problem, yet cross-platform deficiency, high resource usage and time-consuming processes are still valid.

Therefore, this research has tried to address these obstacles and resolve them by proposing a new framework that enhance malware analysis system using cloud computing power.

### 1.3 Research Aim and Objectives

The concentration in this study will be on providing a dynamic analysis as a service which consumes fewer resources, works more efficient and much faster and support a wider range of OSes. So the primary purpose is to create a scalable, fast and efficient cross-platform support for dynamic malware analysis using cloud computing technology to be able to analyze an enormous number of files in less time than before.

The list below shows targets of this research mentioned above:

i. To design and optimize cloud architecture to be able to have more virtual machines on the same physical hardware by considering existing system.

ii. To fine-tune the correlation between computing resources to decrease analyzing time.

iii. To add cross-platform capabilities to the system in order to support analyzing files on different Operating systems such as Android and Windows.

iv. To evaluate the designed system with existing standalone version and previous cloud-enabled system.

## 1.4 Study Scope

The scope of the proposed research is defined in two general parts. The first domain contains extending the engine of analysis to support Android OS as well as Windows. In this case, the system will only generate a report based on the file's behaviour in the Android OS, and it will not conclude that the file is malware or not. Optimizing the infrastructure and enhancing software platform to handle requests faster and more efficient is the second domain of this research.

## 1.5 Thesis Organization

This thesis organized in four main chapter along with first introduction section. Some definition and review about Malware analysis methods, techniques, and existing tools is written in CHAPTER 2. Additionally, some information about virtualization technology, cloud computing paradigm, its characteristics and its components is provided in this chapter. CHAPTER 3 also will contain an explanation about used methodology to state this research objectives with elaborately used configurations. The methodology is explained in this chapter, is used to prove the feasibility of the proposed idea using a different model of experiments. These experiments followed by the results obtained from them will be reported in CHAPTER 4. At the end, conclusion and future work of this research will be presented in CHAPTER 5.

## BIBLIOGRAPHY


[1] K. HALEY, Internet Security Threat Report: Attackers are bigger, bolder, and faster, 2015.

[2] . Symantec, *Internet Security Threat Report 2016,* Retrieved 08/01/2016.

[3] D. Chaffey, *Mobile Marketing Statistics Compilation,* Retrieved 15/03/2016.

[4] M. Egele, T. Scholte, E. Kirda and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," vol. 2, p. 6, 2012.

[5] O. Barakat, S. Hashim, R. Abdullah, A. Ramli, F. Hashim, K. Samsudin and M. Ab, "Malware analysis performance enhancement using cloud computing," vol. 1, pp. 1--10, 2014.

[6] O. Barakat, S. RSA, A. Ramli, F. Hashim, K. Samsudin, I. Al-baltah and M. Al-Habshi, "SCARECROW: Scalable Malware Reporting, Detection and Analysis," vol. 14, p. 1, 2013.

[7] A. Desnos and P. Lantz, *Droidbox: An android application sandbox for dynamic analysis,* 2011.

[8] M. Bierma, E. Gustafson, J. Erickson, D. Fritz and Y. Choe, "Andlantis: large-scale android dynamic analysis," 2014.

[9] O. S. Ugurlu, Stealth sandbox analysis of malware, Bilkent University, 2009.

[10] M. Sikorski and A. Honig, Practical malware analysis: the hands-on guide to dissecting malicious software, no starch press, 2012.

[11] L. Zeltser, Reverse-Engineering Malware: Malware Analysis Tools and Techniques, SANS, 2014.

[12] T. KASAMA, "A Study on Malware Analysis Leveraging Sandbox Evasive Behaviors," 2014.

[13] R. Saradha and S. Education, "Malware Analysis using Profile Hidden Markov Models and Intrusion Detection in a Stream Learning Setting," 2014.

[14] M. Graziano, D. Canali, L. Bilge, A. Lanzi and D. Balzarotti, "Needles in a haystack: mining information from public dynamic analysis sandboxes for malware intelligence," 2015.

[15] D. Octeau, S. Jha, M. Dering, P. McDaniel, A. Bartel, L. Li, J. Klein and Y. Le, "Combining static analysis with probabilistic models to enable market-

scale android inter-component analysis," vol. 1, pp. 469--484, 2016.

[16] J. Svoboda, "Effectively Combining Static Code Analysis and Manual Code Reviews," 2014.

[17] B. Chess and G. McGraw, "Static analysis for security," vol. 6, pp. 76--79, 2004.

[18] D. Pozza, R. Sisto, L. Durante and A. Valenzano, "Comparing lexical analysis tools for buffer overflow detection in network software," in *IEEE*, 2006.

[19] N. Jovanovic, C. Kruegel and E. Kirda, "Pixy: A static analysis tool for detecting web application vulnerabilities," in *IEEE*, 2006.

[20] D. Bruschi, L. Martignoni and M. Monga, "Detecting self-mutating malware using control-flow graph matching," in *Springer*, 2006.

[21] O. Community, *Static Code Analysis,* Retrieved 21/02/2016.

[22] W. Wogerer, "A survey of static program analysis techniques," 2005.

[23] C. Willems, T. Holz and F. Freiling, "Toward automated dynamic malware analysis using cwsandbox," vol. 2, pp. 32--39, 2007.

[24] R. Pirscoveanu, S. Hansen, T. Larsen, M. Stevanovic, J. Pedersen and A. Czech, "Analysis of malware behavior: Type classification using machine learning," in *IEEE*, 2015.

[25] A. Pektaş and T. Acarman, "A dynamic malware analyzer against virtual machine aware malicious software," vol. 12, pp. 2245--2257, 2014.

[26] D.-J. Wu, C.-H. Mao, T.-E. Wei, H.-M. Lee and K.-P. Wu, "Droidmat: Android malware detection through manifest and api calls tracing," in *IEEE*, 2012.

[27] Y. Zhou, Z. Wang, W. Zhou and X. Jiang, "Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets.," 2012.

[28] I. Santos, J. Devesa, F. Brezo, J. Nieves and P. Bringas, "Opem: A static-dynamic approach for machine-learning-based malware detection," in *Springer*, 2013.

[29] L. Weichselbaum, M. Neugschwandtner, M. Lindorfer, Y. Fratantonio, V. Veen and C. Platzer, "Andrubis: Android malware under the magnifying glass," p. 5, 2014.

[30] S. Gadhiya and K. Bhavsar, "Techniques for malware analysis," 2013.

[31] L. Gheorghe, B. Marin, G. Gibson, L. Mogosanu, R. Deaconescu, V.-G. Voiculescu and M. Carabas, "Smart malware detection on Android," vol. 18, pp. 4254--4272, 2015.

[32] A. Apvrille and L. Apvrille, "Sherlockdroid, an inspector for android marketplaces," 2014.

[33] L. Yan and H. Yin, "Droidscope: seamlessly reconstructing the os and dalvik semantic views for dynamic android malware analysis," 2012.

[34] J. Walters, V. Chaudhary, M. Cha, S. Guercio and S. Gallo, "A comparison of virtualization technologies for HPC," in *IEEE*, 2008.

[35] A. Khan, A. Zugenmaier, D. Jurca and W. Kellerer, "Network virtualization: a hypervisor for the Internet?," vol. 1, pp. 136--143, 2012.

[36] M. Jones, "Virtio: An I/O virtualization framework for Linux," 2010.

[37] A.-C. Bujor and R. Dobre, "KVM IO profiling," in *IEEE*, 2013.

[38] D. Sarna, Implementing and developing cloud computing applications, CRC Press, 2010.

[39] . KVM, *KVM FAQ,* Retrieved 18/02/2016.

[40] T. Deshane, Z. Shepherd, J. Matthews, M. Ben-Yehuda, A. Shah and B. Rao, "Quantitative comparison of Xen and KVM," pp. 1--2, 2008.

[41] M. Kebede, Performance comparison of btrfs and ext4 filesystems, 2012.

[42] M. Jones, "Anatomy of ext4-Get to know the fourth extended file system," 2010.

[43] K. Qian, L. Yi and J. Shu, "ThinStore: Out-of-Band Virtualization with Thin Provisioning," in *IEEE*, 2011.

[44] . VMware, *Virtual Disk Provisioning Policies,* Retrieved 19/10/2016.

[45] M. Dutch, "Understanding data deduplication ratios," 2008.

[46] . Wikipedia, *Thin Provisioning,* Retrieved 19/10/2016.

[47] C. Evans, *Write-through, write-around, write-back: Cache explained,* Retrieved 07/2016.

[48] A. Bridgwater, *Treat cloud servers like cattle, not puppies,* Retrieved

23/09/2015.

[49] R. Hirschfeld, *Mayflies and Dinosaurs (extending Puppies and Cattle),* Retrieved 23/09/2016.

[50] R. Hirschfeld, *Research showing that Short Lived Servers ("mayflies") create efficiency at scale,* Retrieved 23/09/2016.

[51] R. Hirschfeld, *Short lived VM (Mayflies) research yields surprising scheduling benefit,* Retrieved 23/09/2016.

[52] M. Koster, *Cloning VMs with KVM,* Retrieved 23/09/2016.

[53] L. Wang, J. Tao, M. Kunze, A. Castellanos, D. Kramer and W. Karl, "Scientific Cloud Computing: Early Definition and Experience.," 2008.

[54] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and . others, "A view of cloud computing," vol. 4, pp. 50--58, 2010.

[55] . Wikipedia, *Cloud Computing,* Retrieved 28/05/2016.

[56] L. Schubert, K. Jeffery and B. Neidecker-Lutz, "The Future of Cloud Computing--Report from the first Cloud Computing Expert Working Group Meeting," 2010.

[57] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.

[58] . Rackspace, *Rackspace: Managed Dedicated & Cloud Computing Services,* Retrieved 2016.

[59] . Amazon, *Amazon Web Services (AWS) - Cloud Computing Services,* Retrieved 2016.

[60] . Google, *App Engine - Platform as a Service | Google Cloud Platform,* Retrieved 2016.

[61] . Microsoft, *Microsoft Azure: Cloud Computing Platform & Services,* Retrieved 2016.

[62] T. Dillon, C. Wu and E. Chang, "Cloud computing: issues and challenges," in *Ieee*, 2010.

[63] S. Goyal, "Public vs private vs hybrid vs community-cloud computing: A critical review," vol. 3, p. 20, 2014.

[64] A. Li, X. Yang, S. Kandula and M. Zhang, "CloudCmp: comparing public

cloud providers," in *ACM*, 2010.

[65] O. Sefraoui, M. Aissaoui and M. Eleuldj, "OpenStack: toward an open-source solution for cloud computing," vol. 3, 2012.

[66] . OpenStack, *OpenStack Architecture,* Retrieved 2015.

[67] R. Kumar, N. Gupta, S. Charu, K. Jain and S. Jangir, "Open source solution for cloud computing platform using OpenStack," vol. 5, pp. 89--98, 2014.

[68] . OpenStack, *Getting Started,* Retrieved 2015.

[69] C. community, *CloudStack's History,* Retrieved 17/03/2016.

[70] R. Kumar, K. Jain, H. Maharwal, N. Jain and A. Dadhich, "Apache cloudstack: Open source infrastructure as a service cloud computing platform," pp. 111--116, 2014.

[71] D. Freet, R. Agrawal, J. Walker and Y. Badr, "Open source cloud management platforms and hypervisor technologies: A review and comparison," in *IEEE*, 2016.

[72] J. Hynninen, "Open source cloud platforms," pp. 53--58, 2013.

[73] L. Martignoni, R. Paleari and D. Bruschi, "A framework for behavior-based malware analysis in the cloud," in *Springer*, 2009.

[74] C. Martinez, G. Echeverri and A. Sanz, "Malware detection based on cloud computing integrating intrusion ontology representation," in *IEEE*, 2010.

[75] . OpenStack, *OpenStackAndItsCLA,* Retrieved 2015.

[76] R. Landmann, D. Cantrell, H. De and J. Masters, "Red Hat Enterprise Linux 6 Installation Guide," pp. 97--101, 2010.

[77] S. Enterprise, *Internet Security Threat Report 2014,* 2015.

[78] C.-W. Huang, M. Chen and D. Zavin, *Android x86-Porting Android to x86,* Available on Dec 2011.

[79] . VirusShare, Retrieved 10/01/2014.

[80] K. Chamarthy, *QCOW2 backing files and overlays,* Retrieved 22/09/2016.

[81] C. Guarnieri, A. Tanasi, J. Bremer and M. Schloesser, *The cuckoo sandbox,* 2012.

[82] C. Wojner, *Mass Malware Analysis: A Do-It-Yourself Kit,* 2009.

[83] . OpenStack, *Storage Decisions,* Retrieved 2015.

[84] B. Leiner, V. Cerf, D. Clark, R. Kahn, L. Kleinrock, D. Lynch, J. Postel, L. Roberts and S. Wolff, "A brief history of the Internet," vol. 5, pp. 22--31, 2009.

[85] P. Innella, "A brief history of network security and the need for adherence to the software process model," pp. 1--15, 2008.

[86] C. Foundation, *Additional Software, Requirements, Cuckoo Sandbox,* Retrieved 2015.