A system for wide detection of collaborative malicious images in the Cloud using machine learning and VMI

Mohammad Reza Memarian
Department of Information Technology,
University of Turku, Finland
moreme@utu.fi

Mauro Conti
Department of Mathematics
University of Padua, Italy
conti@math.unipd.it

Ville Leppanen
Department of Information Technology,
University of Turku, Finland
ville.leppanen@utu.fi

November 24, 2014

Abstract

Abusing cloud services for cyber crime is a prevalent threat to cloud computing. Some of the cloud essential characteristics are main origin of this threat in the cloud. One example of these kind of usages is virtual machines which collaborate together to conduct cyber attacks such as Distributed Denial of Service (DDOS) attack either to a system inside the cloud or outside the cloud that is result of formation of botnet. Various works have been done to develop Intrusion detection systems for cloud. But still the solution for detection of botnets in the cloud is not mature and it is not feasible to have host based IDS on every machine. In this paper we present a system which has a broad view over all images by placing the detecting module in cloud controller level analyzing information gathered in an agent less manner from Virtual machines. The system gathers wide VM's system and network state information. These generic information enables our system to have a broad view over the entire cloud live images to identify malicious collaborative entities. These virtual machines can be located in distributed manner over various clusters in the cloud. Gathered system and network level information can form data sets to be analyzed by machine learning techniques.

1 INTRODUCTION

Cyber criminals are moving toward creating more sophisticated malware as using heavily cryptography in their code and placing their servers in the darknet is getting trendy. As users and firms are moving their data to clouds, attackers are redirecting their focus to cloud too. As described in [1], Botnets are becoming more resilient and responding faster to countermeasures. They integrate multiple backup forms of command and control. Also botnet's trend is moving toward ransomeware attack on user's data in the cloud and banking malware botnet is on the growth. On the other hand, cloud computing provides suitable infrastructure for both acceptable and malicious usages. Based on [2], five essential cloud characteristics are on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service. From the mentioned characteristics, two which empower malicious activities are rapid elasticity and on-demand self-service. On demand self-service refers to ability for users to provision cloud services without human interaction. Rapid elasticity refers to ability of user to shape the consuming cloud resources based on need. So even small companies or individuals can have access to vast amount of computing resources. In [3], Cloud Security Alliance has addressed top nine cloud security threats in year 2013. Abuse of cloud services is one of the threats placed in the mentioned list.

Security is one of the largest obstacles in adopting cloud computing by companies. But the question is, what are the characteristics of cloud-specific vulnerabilities that create cloud-specific threats? As discussed in [4], a vulnerability is applicable to cloud computing if that vulnerability exists in one of the core technologies of cloud computing. As cloud computing is designed on the idea of decreasing cost and increasing efficiency, it is constructed on Virtualization technology. So virtualization technology is one of the core technologies in cloud computing architecture. As a result, vulnerabilities that are threats to virtualization are indeed threats to cloud computing too. So tackling vulnerabilities in virtualization leads to increasing security of the cloud as virtualization security has direct relationship with cloud security. Virtualization vulnerabilities can exist in any layer of Virtualization architecture such as hypervisor or virtual machine layer. Security of images running on virtual machine's structure while holding applications, are critical as they are the the most inner entity in cloud design circle. Securing VM images and monitoring activities of the small entities, boots up overall security of the cloud as images construct base system of the cloud. VM images must have high integrity as they determine initial state of the virtual machines running in the cloud. Users in the cloud can use from shared third party images. An example of these kind of shared image usage model can be a company advertising their application. The software firm configure the application on a VM image and publish that image for use and test of the application by cloud users. Other users can either publish their specific configured image and share it on the cloud either for free or to sell too. Publishing images can be either to an specific group of users or to a public group of users with the aim of users using a homogenises image. This case can be counted as a case which there is a good intention at the back of it. There can be other cases that sharing an image has malicious intention at back of it. Malicious users can configure and publish their image which has malware embedded in it such a backdoor or a rootkit. Adoption of these images multiple times by users in the cloud can lead to creation of malware propagation in the cloud. Cloud service providers do not have strong controls of image sharing as risk of using the shared images should be handled by image's user. As an example Amazon, a cloud service provider [5], stated some security guides to help users to reduce risks of adopting shared images. Users must take of the matter them self.

According to matters discussed above, botnets have easier way to be deployed in the cloud compared to conventional environment. When bots enter a computer system, they should look for the target vulnerability. But in the cloud, Bots depend much less on victim's s system software stack for exploitation. On the other hand ease of image sharing accelerates malware propagation.

In [6], researchers explained risks that face administrators, image publishers and image retrievers in cloud image repositories. An image management system is proposed to control access to images, track source of images, and provide users and administrators with efficient image filters and scanners that detect and repair security violations. The mentioned research depicts the importance of risk reduction of image repositories and risks that can be involved in using cloud shared images. As mentioned earlier cloud is designed on the idea of cost saving and increasing efficiency which leads cloud to homogenization. Specially in the VMs running cloud core services, homogeneous images are desirable. Infection of one of these machines can reveal the vulnerability that exists in other homogeneous virtual machines too that can end up in vast vMs infection. Other scenario regarding this matter can be that one or multiple malicious users rent multiple virtual machines on the cloud. These virtual machines which are controlled from the beginning by predetermined users can do malicious activity in collaboration with each other.

In this paper we review previous work on cloud-botnet detection. Also we present a a work that takes other approach toward cloud-botnet detection. The goal of this research, is not to design a new method to be use in IDS like system. But we present to a system to broadly detect malicious collaborative images in cloud wide manner. As our system gets benefit of looking symptoms leading to botnet instead of botnet behaviour itself, Detection rate increases while signature database size decreases.

The rest of this research paper is structured as following, section 2 provides background to the research area specifically: Botnet, cloud computing and Virtual machine introspection technique. Section 3 focuses on related works and their approach in cloud-botnet detection and forensic virtual machines. Section 4 discusses our approach toward information gathering and detection system. Section 5 discusses implementation details while section 6 describes our system evaluation and last section describes conclusion and our future work direction.

2 BACKGROUND

This section presents a surface overview of botnets, cloud computing, Virtual machine introspection, detection algorithm to be used and the relationship between them.

2.1 Botnet

Description of Botnet especially in the cloud.

2.2 Cloud Computing

Based on NIST [2], Cloud computing is a model for enabling on-demand access to a shared pool of configurable computing resources which can quickly be employed with minimal management effort and human interaction. Indeed Cloud computing is changing the way that computing services are offered. Cloud computing has 5 essential characteristics which are: on-demand self-service, Broad network access, Resource pooling, Rapid elasticity and measured services. Cloud services are offered through 3 service models which are: Software as a service (SaaS), Platform as a service (PaaS) and Infrastructure as a service (IaaS). Also cloud computing is deployed based on 4 deployment model. The choice of deployment model depends on the environment in which cloud is going to be deployed in. The 4 cloud deployment models are: private cloud, community cloud, public cloud and hybrid cloud.

2.3 Virtual Machine Introspection (VMI)

VMI is a technique used in virtualization enabling VM's system state monitoring from a secure place in agent less manner. Traditional monitoring methods require monitoring system's agent to be placed in the monitored VM. In the case that VM is compromised, the malware on the system can deactivate the agent on the monitored system. VMI tackles the problem of agent oriented techniques in non virtual environments. The monitoring can be done from a secure place like virtual machine monitor (VMM) or any other privileged virtual machine. Garfinkel and Rosenblum in [7], presented an intrusion detection system design based on VMI technique. Various other works are done around this area to enable monitoring and detection system to use power of VMI in their solutions. (Some examples of these kind of papers to be mentioned)

2.4 Machine learning approach for detection

Similarity detection algorithms of machine learning seems to be good. but similarity algorithms is a supervised model that other papers researcher used too. If we can use from unsupervised algorithm that the system detects itself without us giving it some samples of bad malicious behaviour, I think is better.

3 RELATED WORK

In this section we review previous works done on botnet detection in context of cloud and also matter of forensic virtual machine which is quite a new term.

3.1 Cloud-Botnet detection

Researchers in [8], implemented a passive and an active malware detection module in VMM to actively look for information in the VM without installing agent (By using VMI). The solution presented in their work mainly focuses on function in one host and detects zombie machines, based on trained node about bot behaviours. The solution does not have a wide view over the cloud, and it mainly makes the botnet profile based on just the API calls done by the applications and it does not have an approach toward having general detection approach for detecting various botnets as looking for symptoms can extremely help in this area.
A research paper by Kebande and Venter [9], proposed botnet detection methods in the cloud environment using Artificial Immune System (AIS). This mechanism uses negative selection algorithm to match whether the botnet belongs to self or non-self pattern. it gets done by training some detectors on identifying malicious activity pattern. In the time of attack (when bots are attacking to zombies), AIS is trained to detect a malicious activity pattern and observes the behaviour based on the network traffic movement that makes the solution not to be active before the actual action. This is because zombies being member of same botnet exhibit similar characteristics. The proposed detection method does not presents how and where in the cloud this solution must be placed and implemented. Also simplicity which is rule of thumb in security is not followed as the solution is complicated and not clear.
Beidermann and Katzenbeisser in [10] presented a research work to detect computer worm in cloud based on the spreading behaviour. the solution looks for inconsistency in behaviour of machines. Their system only looks for two information that are: start of a new process which is not listed in the white list and loading of a module which is not in the white list. Each process may run by any machines and may not be malware and only start of a process can not indicate malicious activity. Thats why that our system filters the information several times against several parameters to obtain the most accurate result. In this solution also monitoring stages are mentioned that have a randomly look into VMs which is not suitable for continues monitoring. The information obtained from each VM using VMI is sent to a central spreading monitor in the host server That compares the lists. As mentioned, even if two VM on one host have the same black list process, it is not accurate to identify them as zombies of the same botnet.
Watson in [11] presented a distributed detection system that combat malware in multi-server cloud. The research proposed having agents in each VMM of servers that communicate with each other and pass obtained information to each other. Without a central decisioning and information processing point, solution will not be effective.

3.2 Forensic Virtual Machine (FVM)

Researchers in [12], presented a method of detecting malware by identifying the symptoms of malicious behaviour instead of looking for malware itself. This framework presents detection using small independent privileged VMs called forensic virtual machines (FVM) to inspect memory page of other virtual machines using VMI. Each of these FVMs in each host only look for one symptoms. Researchers in [13], Implemented FVMs using MiniOS. MiniOS is a small operating systems distributed by Xen source code and is intended to be used for dom0 disaggregation.

4 APPROACH

In this Section, we present our approach toward information gathering, analysis and processing of obtained data about VM's system state.

4.1 Data gathering methods at each host

System level information are gathered using VMI technique. Introspection technique can give a reach information at any given point about system-level of the each live virtual machine. As described in [12], we assign one FVM per vm to retrieve system level information from each vm's memory page at any given

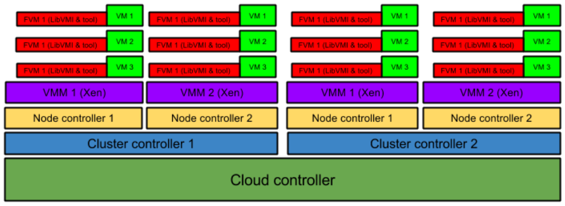

Figure 1: Typical IaaS design with FVM incorporated in it

time. Assigning one FVM per VM as opposed to [13], has the benefit of monitored VM's Trusted computing base (TCB) reduction as each monitored VM must trust to only one FVM assigned to it not the whole FVMs in the host. Also all the symptoms are checked at one place and frequently instead of random check which is not feasible.

4.2 Categorization of obtained data

Information gathered at each host is places in relevant data structure and they are sent to analytic module. Analytic module is placed in cloud controller level that is the highest level in cloud architecture. So hypervisor does not have any role in analysing data. The role of hypervisor is just to pass receive the information from FVMs and pass them to Node controller. Node controller passes them to cluster controller and cluster controller pass the information to cloud controller. Then data and symptoms detected are categorized at cloud controller level.

4.3 Detective algorithm and method

As machine learning is the prevalent method used in these kind of situations, we have used from supervised method of machine learning to detect similar images cloud wide. The similarity checking and filtering is done several times against different parameters to obtain the most accurate list of possible collaborative machines.

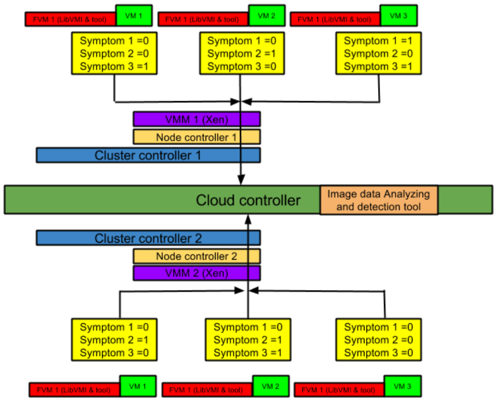

Figure 2: Transfer of gathered data to CLC

5 IMPLEMENTATION

In this section, an overview of implemented system is described.

5.1 xen

We used from Xen as hypervisor for our implementation. Ubuntu 14.4 is our choice for Dom0 Implementation. The reason for using xen is that Xen is a widely used open source Virtual machine monitor. Large cloud service providers like Amazon use Xen as the hypervisor for virtualization infrastructure. Virtual machines running on Xen are referred as domains. There are two domain types in Xen, Privileged domain and unprivileged domain. A supervisor like VM called dom0 runs as privileged domain. It is the first VM that boots after hypervisor starts and holds hardware drivers and control software stack. Other guest virtual machines run in unprivileged mode. Virtual machines running on Xen should run on one of the two modes. The two modes are hardware virtualization (HVM) and Para virtualization (PV). In HVM, Virtual machine is not aware that it is running in a virtualized environment. Whereas in PV mode, Virtual machine experiences some modification and is aware that is running on virtual platform. Virtual machines running in PV mode experience faster system performance. Closed source operating systems like Microsoft Windows, must run is HVM mode because modification of it, is not possible.

5.2 libvmi

We used from LibVMI version 4.4 to conduct obtain information from VM's memory. Libvmi [14] is an introspection technique library, written in C and is capable of reading and writing memory from virtual machines. Libvmi supports Virtual machines running on Xen and KVM hypervisours.

5.3 openstack

Openstack is an open source cloud computing platform suitable for providing IaaS cloud. It is widely used and has a good integration with ubuntu and xen. (more details to be added)

5.4 System implementation detail

As implementation of this solution seems to be tricky, we may do this, Implement the openstack, create two clusters in each one one server, on each server one or two VM, run different related processes on them. on the other hand write a python script to run in Cloud controller to receive these info and using similarity algorithm check the similarity of them. This is the simplest method that comes to my mind, If you have simpler way, it is appreciated to have the idea.

6 Evaluation and Results

Possible evaluation can be that detection rate is good and fast, by showing some graphs like how fast our system can detect the images. how many of these collaborative images should start to attack until they are detected. Maybe we can come up with some evaluations as we start to implement some solutions later.

7 CONCLUSION AND FUTURE WORK

As Cloud is attracting attention from firms to move their data to it, cyber criminals are also moving the their attacks toward cloud as Data is found there. Having monitoring systems that are capable of widely and actively monitor, log and report malicious activities and movements is essential. In this paper we presented a system which acts as super system looking into its subsystems strongly resulting is relating events occurring in each cluster and detect collaborative malicious images. Privacy of data obtained from guest Virtual machines using VMI is a very important point. In the matter of VMI, Privacy of user's activities is overlooked in favour of secure monitoring. Our future research direction will be on privacy of user data in monitoring systems using VMI.

References

[1] SOPHOS, "security threat report 2014," Sophos, Tech. Rep., .

[2] NIST, "Defenition of cloud computing," NIST, Tech. Rep., 2011.

[3] CSA, "Nine threats to cloud," Cloud Security Aliance, Tech. Rep., 2013.

[4] E. s. Bernd Grobauer, ToBias Walloschek, "Understanding cloud computing vulnerabilities," *IEEE Security and Privacy*, 2012.

[5] . () Building amis for aws marketplace. [Online]. Available: https://aws.amazon.com/marketplace/help

[6] G. A. V. B. P. N. Jinpeng Wei, Xiaolan Zhang, "Managing security of virtual machine images in a cloud environment," in *CCSW 2009*.

[7] M. R. Tal Garfinkel, "A virtual machine introspection based architecture for intrusion detection," in *NDSS 2003*.

[8] Y. S. S. M. C. C. Shun-Wen Hsiao, Yi-Ning Chen, “A cooperative botnet profiling and detection in virtualized environment,” in *2013 IEEE Conference on Communication and Network Security (CNS).*

[9] H. Victor .R. Kebande, “A congnitive aproach for botnet detection using artificial immune system in the cloud,” in *2014 IEEE Third International conference on cyber security, cyber warfare and digital forensics.*

[10] S. K. Sebastian Biedermann, “Detecting computer worms in the cloud,” in *Open Problems in Network Security - IFIP WG 11.4 International Workshop, iNetSec 2011.*

[11] M. R. Watson, “Malware detection in the context of cloud computing,” in *2012 PGNet.*

[12] S. T. A. C. I. A. N. Keith Harrison, Behzad Bordbar, “A framework for detecting malware in cloud by identifying symptoms,” in *2012 IEEE 16th international enterprise distributed object computing conference.*

[13] J. S. K. H. C. D. Adrian L. Shaw, Behzad Bordbar, “Forensic virtual machine: Dynamic defence in the cloud via introspection,” in *2014 IEEE International Conference on Cloud Engineering.*

[14] () vmitools. [Online]. Available: https://code.google.com/p/vmitools/