A system for wide detection of collaborative malicious images forming botnets in the cloud by analyzing system level information

Mohammad Reza Memarian
University of Turku
FI-20014
Turku, Finland
moreme@utu.fi

Mauro Conti
University of Padova
Via 8 Febbraio, 2 - 35122
Padua, Italy
conti@math.unipd.it

Ville Leppanen
University of Turku
FI-20014
Turku, Finland
ville.leppanen@utu.fi

ABSTRACT

Abusing cloud services for cyber crime intentions is a prevalent threat to cloud computing technology. Some of the cloud essential characteristics are main origin of the mentioned threats to the cloud. One example of these kind of usages is infected virtual machines forming botnets to conduct cyber attacks such as Distributed Denial of Service (DDOS) either against a system inside the cloud or a system outside of the cloud. Various works have been done to develop Intrusion detection systems for cloud to mitigate threats like DDOS. But still the solution for detection of botnets in the cloud are not mature as DDOS attacks are on rise. On the other hand 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 of cloud platform coordinating and analyzing information gathered in an agentless manner from Virtual machines using Virtual Machine Introspection (VMI). The system gathers wide VM's system state information. The generic information gathered enables our system to have a broad view over the entire cloud live images and detect malicious collaborative entities. These virtual machines can be located in distributed manner over various clusters in the cloud. As nature of cloud is distributed, cloud monitoring systems should be distributed too. Gathered system level information can form data sets to be analyzed by the analyzing module using machine learning techniques.

Keywords

Cloud computing security, Botnet detection, Virtual machine introspection

Categories and Subject Descriptors

C.2 [**COMPUTER-COMMUNICATION NETWORKS**]: Security and protection

General Terms

Security

1. INTRODUCTION

Cyber criminals are moving toward creating more sophisticated malware for creating undetectable botnets. Based on [1], DDOS attacks which are one of the consequences of botnet formation, tend to change their attack vector for remaining undetectable instead of relying on single vector in one attack. On the other hand, network protocols design form a big portion of the problem as they were not designed by security in mind manner. From time to time vulnerabilities in them get exploited which cause massive damages to systems. One type of misuse from cloud services can be botnet formation. DDOS is one of the consequences of botnet formation. There are various DDOS attacks which threaten cloud security. One of the DDOS attacks that is an on rise prevalent attack vector and is result of vulnerabilities in network protocols such as NTP, DNS and SSDP is UDP reflection flood attack. These types of attacks are sort of easy attacks while having large impact on the target. As users and firms are moving their data to clouds, attackers are redirecting their attack focus to cloud too. As described in [2], Botnets are becoming more resilient and responding faster to countermeasures. They integrate multiple backup forms of command and control. On the other hand, cloud computing provides suitable infrastructure for both acceptable and malicious usages.

Based on [3], 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. Based on [4], Using online DDOS services does not need sophisticated knowledge and they can be rented by as little cost as 5$. So even small companies or individuals can have access to vast amount of computing resources in a short period of time with low financial requirements. Cloud security alliance (CSA) has addressed top nine cloud security threats in year 2013. Abuse of cloud services is one of the threats placed in the mentioned list as well as Distributed denial of

service attacks [5].
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 [6], 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 an example of these kind of shared image usage model can be a company advertising their application. The software firms 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 to other too. images can be published either to a 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 across the cloud can lead to malware propagation across the cloud. Cloud service providers do not have strong controls over image sharing as risk of using the shared images should be handled by image users. As an example Amazon, a cloud service provider, stated some security guides to help users to reduce risks of adopting shared images that depicts users must handle the risk in this case [7].

According to matters discussed above, creating network of collaborative images or botnets formation are easier in the cloud compared to conventional environments. When a bot enters a computer system, it should look for some distinctive vulnerabilities. But in the cloud, Bots depend much less on victim's system software stack for exploitation. Ease of image sharing and interest for employing homogeneous images are factors that accelerate malware propagation. Furthermore Infection methods may differ in a cloud botnet to mislead detection systems. While in the previous works, most of the focus has been on one way infection method.

In [8], researchers explained risks that administrators, image publishers and image retrievers face 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 shared cloud images. As mentioned earlier cloud is designed on the idea of cost saving and increasing efficiency which leads cloud to homogenization. Specially in the case of 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. It can end up in vast systems infection across the cloud. 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 specific users can do malicious activities in collaboration with each other.

As threats regarding DDOS attacks which are result of botnet creation are on rise, lack of a system that has a wide look over the cloud can correlate all the information and coordinate them is felt. In this paper we reviewed previous works done on cloud-botnet detection. Then we present a system that takes different approach for botnet detection specifically in the cloud by getting advantage of one of the cloud core technologies that is virtualization. The goal of this research is not to design a new method to be used in IDS like systems or integrate conventional IDS systems with cloud platform. But we present a system to broadly detect malicious collaborative images in cloud wide manner. As our system gets benefit of looking for 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 study about botnet, cloud computing, virtual machine introspection, detection algorithm to be used and the relationship between them.

2.1 Botnet

To Be Added.

2.2 Cloud Computing

Based on national institute of standards and technology (NIST), 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 [3]. Indeed Cloud computing is changing the way that computing services are offered. As

mentioned earlier, 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 models. 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. Security issues are one of the biggest obstacles in adoption of cloud through companies.

2.3 Virtual Machine Introspection (VMI)

VMI is a technique used in virtualization enabling VM's system state monitoring from a secure place in an agentless 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 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 [9], presented an intrusion detection system design based on VMI technique. Various other works are done around this area to enable monitoring and detection systems to use power of VMI technique in their solutions. (Some examples of these kind of papers to be mentioned)

2.4 Machine learning approach for detection

To be investigated during the time until next skype call on 17.12.14. Similarity detection algorithms of machine learning seem to be good. But similarity algorithms are supervised model that other 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. Generally It should be investigated and studied more, the results will be reported.

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 [10], 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 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 involve entites. A research paper by Kebande and Venter [11], 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. Beidermann and Katzenbeisser in [12] presented a research work to detect computer worms 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 predetermined white list and loading of a module which is not in the predetermined white list. Generally start of a black list process even on several machines can not necessarily indicate malicious activity. 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. Watson in [13] presented a distributed detection system that combat malware in multi-server cloud. The research proposed having agents in each hypervisor of servers that communicate with each other and pass obtained information to each other. Without a central decisioning and information processing point, solutions will not be effective.

3.2 Forensic Virtual Machine (FVM)

Researchers in [14], 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 [15], 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.

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 in time about system-level of each live virtual machine. Example of these informations can be list of running processes, loaded modules and opened files. As shown in figure 1, we assign one FVM per vm to retrieve system level information from each vm's memory page at any given time. Assigning one FVM per VM as opposed to [15], 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. Also in the time of FVM infection, other monitored machines can remain immune from consequences of one of the FVM's infection.

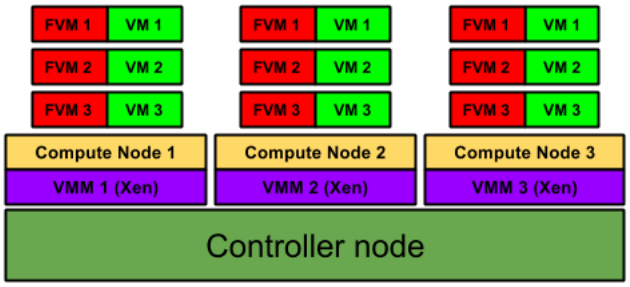

Figure 1: Typical IaaS design with FVM incorporated in it

4.2 Categorization of obtained data

Information gathered at each host is gets in relevant data structure and they are sent to analytic module. As shown in figure 2, analytic module is placed in controller node level that is the highest level in cloud architecture. So in our design as opposed to previous works, VMM does not have any role in analyzing data that results in offloading this task from VMM. Informations gathered by FVMs are passed to compute node. Then Compute node of each host pass them to the cloud controller node. Then data and symptoms detected are analyzed and proceeded at controller node level.

4.3 Detective algorithm and method

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

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 hypervisor. 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 in privileged mode. 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

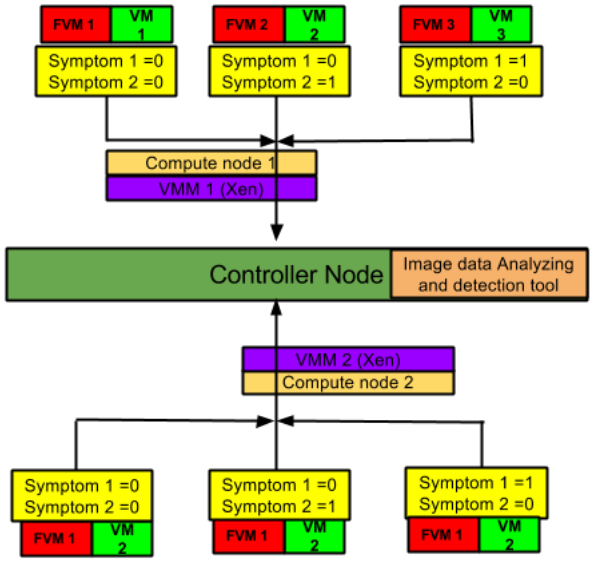

Figure 2: Transfer of gathered data to controller node

mode because modification of it, is not possible. Openstack entities run in unprivileged mode in xen.

5.2 Libvmi

We used from LibVMI version 4.4 to conduct obtain information from VM's memory. Libvmi [16] 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 hypervisors.

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 discussed in Skype call on 08.12.14, development of data gathering modules using LibVMI has been decided to be started. Although our solutions focuses on assigning one FVM per VM but in our development of this code, we can put this module in Dom0. Because at this level it is important for us to develop the module itself and data that can be gathered.

6. EVALUATION AND RESULTS

Possible evaluation can be that detection rate is good compared to some statistics and fast, by showing some graphs like how fast and accurate our system can detect the images. How many of these collaborative images should start to attack until their network is 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, cybercriminals 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 in 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.

8. REFERENCES

[1] âĂć, "Verisign distributed denial of service trends report, 2nd quarter 2014," Versign, Tech. Rep., âĂć.

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

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

[4] C. Wueest, "The continued rise of ddos attacks," Symantec, Tech. Rep., âĂć.

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

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

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

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

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

[10] 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)*.

[11] 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*.

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

[13] M. R. Watson, "Malware detection in the context of cloud computing," in *2012 PGNet*.

[14] 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*.

[15] 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*.

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