A framework for distributed detection of infected machines in cloud

Position paper

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

Mauro Conti
Department of mathematics
University of Padova
conti@math.unipd.it

Ville Leppanen
Department of Information Technology
University of Turku
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 threat to the cloud. One example of these kind of malicious usages is infected virtual machines forming botnets to conduct cyber-attacks such as Distributed Denial of Service (DDOS). DDOS can happen either against a system inside the cloud or a system outside of the cloud using cloud resources. Various works have been done to develop Intrusion detection systems (IDS) for cloud to mitigate threats like DDOS. But still the solutions for detection of botnets in the cloud are not mature as DDOS attacks which are one of the consequences of formation of botnets, are on rise. Most of the solutions are designed for being used on a single host while it is not feasible to have host based IDS on every machine. In this paper we present an aproach which has a broad view over all virtual instances in the cloud by placing the detecting module in a higher level than host level in the cloud. Detection module coordinates and analyzes information gathered in an agent-less manner from Virtual machines (VM) using Virtual Machine Introspection (VMI). The system gathers wide VM's system state information. The generic information gathered enables detection module to have a broad view over the entire cloud live images and detect malicious collaborative entities. As nature of cloud, VMs in cloud are distributed over various clusters. So cloud monitoring systems should be distributed too.

1. INTRODUCTION

Cyber criminals are moving toward creating more sophisticated malware for creating undetectable botnets. Based on [12], 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, design of network protocols form a portion of the problem as some of them were not designed by security in mind manner. From time to time vulnerabilities in network protocols get exploited which may result in massive attacks and damage systems. One type of misuse from cloud services can be botnet formation. As mentioned earlier, conducting DDOS attack is one of the intensions of botnet controllers. 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 to conduct attacks while having large impact on the target. Normally reflection attacks are hard to be detected as the preliminary request is a legitimate request with spoofed IP address of the victim. In These attacks, reflector which can be a vulnerable DNS or NTP server, has a much larger respond to a query sent by a client. The amount of excess forwarded traffic is called amplification factor. NTP amplification factor can be up to 556.9 meaning that the attacker can generate 556.9 times more traffic than the request that is sent [4]. Detection of these attacks are not easy as the request looks legitimate to reflector. As users and firms are moving their data to clouds, attackers are redirecting their attack focus to cloud too. As described in [10], 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 [8], 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 users to shape the consuming cloud resources based on their need. Using online DDOS services does not need sophisticated knowledge and also they can be rented by as little cost as 5$ [11]. As a result 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 [9]. Abuse of cloud services is one of the threats placed in the mentioned list as well as Distributed denial of service at-

tacks.
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? A vulnerability is applicable to cloud computing if that vulnerability exists in one of the core technologies of cloud computing [13]. As cloud computing is designed on the idea of reducing cost and increasing efficiency, it is constructed on virtualization technology. So virtualization technology is one of the core technologies in cloud computing architecture. Vulnerabilities that are threats to virtualization are indeed are threats to cloud computing too. Tackling vulnerabilities in virtualization leads to increasing security of the cloud as virtualization security has direct relationship with cloud security. Security of images running on virtual machine's structure in the cloud while holding applications, are critical to the cloud security in general as they are the the most inner entity in cloud design virtualization circle. Securing VM images and monitoring activities of the these 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. Distribution of these images may require some cautions as users in the cloud can use from shared third party images. Example of these kind of shared image usage model can be a company advertising its 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 others too. Shared 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 homogeneous image. This case can be counted as a case which there is a good intention at the back of it. But there can be other cases that sharing an image has malicious intention at back of it. Malicious users can configure and publish infected image which may have 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 cloud wide. While usage of shared images can have a high risk, 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. It depicts users must handle the risk in this case [2].
According to matters discussed above, creating network of collaborative images or botnet 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 (specifically in the case of image sharing), bots depend much less on victim's system software stack for exploitation. Ease of image sharing and interest for employing homogeneous images by group of users and cloud service providers (CSP) 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 [22], 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 reducing cost and increasing efficiency which leads cloud to homogenization. Specially in the case of VMs running cloud core services, homogeneous systems 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 DDOS attacks and in general botnet related attacks are on rise in the cloud, presence of a system that has a wide look over the cloud and is able to correlate and coordinate all the information cloud wide is vital. In this paper we reviewed previous works done on cloud-botnet detection. Then we present a system design that adopts a centralized approach for botnet detection in the cloud by getting advantage virtualization as one of the cloud core technologies. The goal of this research is to 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 behavior 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 technical approach toward information gathering and detection system design. Section 5 discusses implementation and simulation details and section 6 which is the last section discribes conclusion and our future work direction.

2. BACKGROUND

This section presents a surface study about botnet, cloud computing and virtual machine introspection technique.

2.1 Botnet

Botnet is a group of infected computer systems that enable a controller to have control over the system. The infected entities are called zombies and in most cases, users of victim machines are not aware that their system is compromised. There are various malicious usages of botnets for controllers. These usages are conducting DDOS attacks, spamming, search engine optimization poisoning, pay-per-click fraud, financial fraud, bitcoin mining and information stealing [1]. There are various ways that victim systems are infected such as through opening an email with malicious content, installing pirated software and visiting malicious websites. When systems are infected, malware installs a backdoor to enable botnet controller to have consistent communication with zombies. Infected systems may send primary vicitm system's information such as IP address, Geographical location, operating system type, host name as other information which helps classifying zombies. From

time to time malware may download newer version of itself to get updated. Infected machines and bot controller may communicate with each other through various ways such as P2P protocol, HTTP commands and IRC channels. Nowadays, cost and skills of having a botnet have been much reduced. Botnets source code can be found in under ground black market websites with very low cost while they are designed to be easily worked.

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. These resources are able to quickly be employed with minimal management effort and human interaction [8]. Indeed Cloud computing is changing the way that computing services are offered. 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 on. The 4 cloud deployment models are: private cloud, community cloud, public cloud and hybrid cloud. As mentioned earlier, 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 technology enabling VM's system state monitoring from a secure place outside of the monitored VM. VMI is done is an agent-less manner. Traditional monitoring methods require monitoring 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 [21], presented an intrusion detection system design based on VMI technique.

3. RELATED WORK

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

3.1 Cloud-Botnet detection

Researchers in [15], 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 functions in one host and detects zombie machines, based on trained node about bot behaviors. 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 involved entities. A research paper by Kebande and Venter [16], 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 activities pattern. In the time of attack (when bots are attacking to zombies), AIS is trained to detect a malicious activity pattern and observes the behavior based on the network traffic movement. Beidermann and Katzenbeisser in [19] presented a research work to detect computer worms in the cloud based on the spreading behavior. The solution looks for inconsistency in behavior of machines. Their system is limited to only looks for two information that are in systems 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 VMM layer that compares the lists. Although it is mentioned that the system is able to detect various unknown malware but trojans today do not necessarily start a new process with an undefined name. Watson in [17] 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)

Due to modular design trend of malware, researchers in [14], presented a method of detecting malware by identifying the symptoms of malicious behavior 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 [20], 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. TECHNICHAL APPROACH

In this Section, we present our approach toward information gathering, analysis and processing of obtained data.

4.1 Data gathering method at each host

System level information are gathered using VMI technique. Introspection technique can give a set of reach system-level information at any given point in time about each live virtual machine. Example of these informations can be list of running processes, loaded modules, opened files and running services. As shown in 1, in our design 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 [20], 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 once and frequently instead of random check which is not feasible in term of detection. Also in time of FVM infection, other monitored machines can remain immune from consequences of one of the FVM's infection.

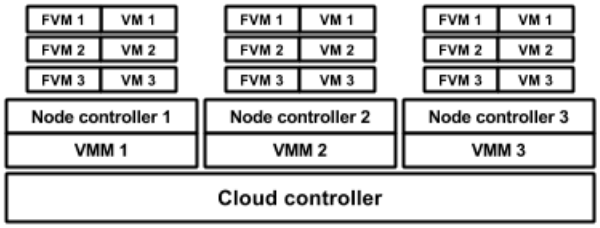

Figure 1: Typical IaaS design with FVM incorporated in it

As mentioned earlier and shown in Figure 3, forensic virtual machines look for symptoms of malicious activities. These symptoms should be defined by determining inconsistencies in VMs as our approach focuses on system level inconsistencies. Based on malware analysis information available on Symantec website [3], we constructed a table depicting common modifications that 9 botnet intended malware do to victim system after infecting them. As mentioned earlier, the focus of our framework is to detect the infected colloborative virtual machines based on the similar malicious symptoms. Based on table 1 and [18], inconsistencies that we have considered them to be prevalent among botnet intended malicious codes are: File/Folder creation registry key creation, existence of hidden (unlinked) processes, existence of hidden (unlinked) DLL and existence of unnormal loaded service in memory.

4.2 Categorization of obtained data

Information gathered at each host gets in relevant data structure by forensic virtual machine and they are sent to analytic module. As shown in 3, analytic module is placed in cloud controller level that is the the most outer level in cloud platform architecture. So in our design as opposed to previous works, VMM does not have any role in analyzing data which results in offloading this task from VMM. Informations gathered by FVMs are passed to node controller. Then node controller of each host passes them to the cloud controller node. Data and symptoms detected are analyzed and proceeded at cloud controller level. Architecture of detection system and relationship among envolved modules are depicted in 2.

5. IMPLEMENTATION AND SIMULATION

In this section, an overview of configured system for test is described.

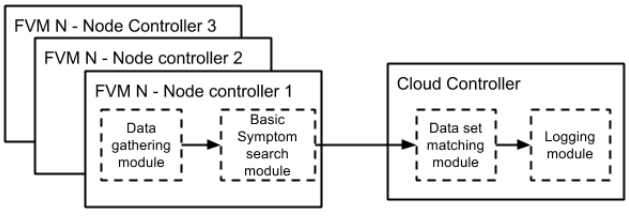

Figure 2: Architecture of detection system

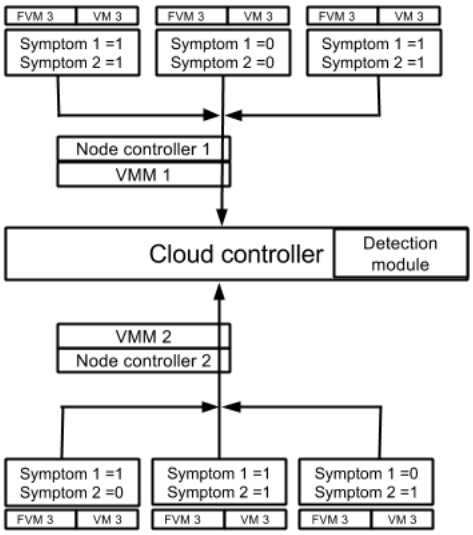

Figure 3: Transfer of gathered data to controller node

5.1 XEN

We used Xen as VMM for our system configuration. Ubuntu 14.4 (Linux kernel 3.13.0-24-generic) is our choice for Dom0 Implementation. The reason for using XEN is that while XEN is a widely used VMM, it is an open source virtualization solution. Large cloud service providers like Amazon use Xen as the VMM for virtualization infrastructure [7]. 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 VMM starts and holds hardware drivers and control software stack. Other guest virtual machines run in unprivileged mode. Dom0 is allowed to have access to memory pages of other virtual machines so VMI is possible from Dom0 level. By modifying Xen access control, other VMs can be granted access to introspect other VMs too. Virtual machines running on Xen should run on one of the two available 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 them, is not possible. We run a guest operating system running in HVM mode while having Microsoft windows 7 as operating system.

5.2 Introspection tools

We used LibVMI version 4.4 to conduct VMI for obtaining information from VM's memory. Libvmi is an introspection library, written in C and is capable of reading and writing memory from virtual machines [5]. Libvmi supports Virtual machines running on Xen and KVM VMM. In conjunction with Libvmi library, we used Volatility [6]. Volatility is an advanced open source memory forensic framework which of-

System modification	Rustock.B (2006)	Virut (2007)	Koobface (2008)	Tidserv (2008)	Qakbot (2009)	Pilleuz (2009)	Zbot (2010)	Carberp.B (2014)
File/Folder created	✓		✓	✓	✓	✓	✓	✓
Files/Folder deleted				✓		✓		
Files/Folder modified		✓		✓	✓	✓		
Registry entry created	✓	✓	✓	✓	✓	✓	✓	✓
Registry entry deleted								
Registry entry modified								
Kernel Modification	✓							
Code injection into process/DLL/App	✓	✓	✓	✓	✓	✓		

Table 1: Botnet intended malware functionalities

fers variety of functions to users for extraction of data from virtual machine snapshots as well as memory dumps. As Libvmi has integrity with Volatility, it is possible to used Volatility functions on live VMs while using Libvmi.

5.3 Simulation

We setup an environment for simulation and ran a guest VMs (domU) to get it infected with a malware. In this test we infected domU with an instance of Koobface. Libvmi and Volatility are configured to gather information from Dom0. We used from psscan plugin of volatility to gather all running processes including hidedn (Unlinked) processes that are not shown in the task list of the operating system and terminated processes which with random introspection technique is not detectable. When domU became infected, a process with name bolivar30.exe started. Then immidiately dllhost.exe and regedit.exe were started and terminate. They were shown in by psscan plugin as terminated process. As a result of regedit.exe start and termination, a registry entry was created to start the malicious process each time the system starts. So in this case our framework looks into the registry key which is prevelant for creation of value to check whether any value is added to the key or not. It does not check what values are added. Ofcourse adding value to a registry key can be done for legitimate purposes too but existence of this symptoms in conjuction with others in a cloud wide manner can be a brief indication of wide cloud infection. For the system to be able to distingush among different infected groups, variety of different symptoms must be checked.

6. CONCLUSION AND FUTURE WORK

Abusing cloud services is a prevalent threat toward cloud computing security. The need for having distributed and efficent monitoring systems is vital to empower security of the cloud. By making sure about security of the most inner entity of the cloud computing which are virtual machine images and VMs, the entire security of the cloud can be increased. Having distributed 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 framework which is able to act as a super system looking into its subsystems and strongly resulting in relating events occurring in each cluster and detect collaborative running malicious images.

There are two future direction which we would like to follow as our future work. First we would like to move more toward depicting detection ratio of our framework by implementing the proposed framework in different cloud scenarios. Second is enhancing privacy of data obtained from guest VMs using VMI as is a very important point. In the researches which VMI is involved, privacy of user's activities is generally overlooked in favor of secure monitoring. VMI technique can reveal various private information from inside the guest VMs. These information can be target of misuse by malicious insiders. So our second future research direction will be on privacy of user data in monitoring systems using VMI.

7. REFERENCES

[1] Anatomy of a Botnet. Technical report, Fortinet.

[2] Building AMIs for AWS Marketplace. `https://aws.amazon.com/marketplace/help`.

[3] Symantec website. www.symantec.com.

[4] United States computer emergency readiness team. www.us-cert.gov.

[5] VMItools. https://code.google.com/p/vmitools/.

[6] Volatility foundation website. www.volatilityfoundation.org.

[7] XEN project website. www.xenproject.org.

[8] Defenition of cloud computing. Technical report, National Institute of Standards and Technology (NIST), 2011.

[9] Nine threats to cloud. Technical report, Cloud Security Aliance(CSA), 2013.

[10] Security threat report 2014. Technical report, Sophos, 2014.

[11] The continued rise of DDoS attacks. Technical report, Symantec, 2014.

[12] Verisign distributed denial of service trends report, 2nd quarter 2014. Technical report, Versign, 2014.

[13] Grobauer, B. and Walloschek, T. and Stocker, E. Understanding Cloud Computing Vulnerabilities. *Security Privacy, IEEE*, 9(2):50–57, 2011.

[14] K. Harrison, B. Bordbar, S. Ali, C. Dalton, and A. Norman. A framework for detecting malware in cloud by identifying symptoms. In *Proceedings of the IEEE 16th International Enterprise Distributed Object Computing Conference, 2012*, EDOC, pages 164–172, 2012.

[15] S.-W. Hsiao, Y.-N. Chen, Y. Sun, and M. C. Chen. A cooperative botnet profiling and detection in virtualized environment. In *In proceedings 2013 IEEE Conference on Communications and Network Security (CNS)*, pages 154–162, 2013.

[16] V. Kebande and H. Venter. A cognitive approach for botnet detection using Artificial Immune System in the cloud. In *Proceedings of 2014 Third International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, pages 52–57, 2014.

[17] M. R. Watson. Malware detection in the context of cloud computing. In *Proceedings of 13th annual post graduate symposium on the convergence of telecommunication, networking and broadcasting*, 2012.

[18] A. H. Michael Sikorski. *Practical malware analysis*.

[19] S. K. Sebastian Biedermann. Detecting Computer Worms in the Cloud. In *Proceedings of iNetSec 2011, Open Problems in Network Security - IFIP WG 11.4 International Workshop*.

[20] Shaw, A.L. and Bordbar, B. and Saxon, J. and Harrison, K. and Dalton, C.I. Forensic Virtual Machines: Dynamic Defence in the Cloud via Introspection. In *Proceedings of 2014 IEEE International Conference on Cloud Engineering (IC2E)*, pages 303–310, 2014.

[21] Tal Garfinkel, Mendel Rosenblum. A virtual machine introspection based architecture for intrusion detection. In *Proceedings of NDSS Symposium 2003*, 2003.

[22] Wei, Jinpeng and Zhang, Xiaolan and Ammons, Glenn and Bala, Vasanth and Ning, Peng. Managing Security of Virtual Machine Images in a Cloud Environment. In *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, pages 91–96, 2009.