Security against Side Channel Attack in Cloud Computing

Bhrugu Sevak

Abstract- Cloud computing is a word that delivering hosted service over the internet. Cloud computing has been ideate as the next generation architecture of IT enterprise because of it's provides ubiquitous network, cost reducing, flexibility and scalability to users. Now days with the fast growing of cloud computing technology introduces new more vulnerabilities so security is considered to be one of the most critical aspect in clod computing environment due to the confidential and important information stored in the cloud. As per AMAZONE EC2 service case study it is possible to identify the particular target VM(virtual machine) in internal cloud infrastructure and then placed new VM with targeted VM and extract confidential information from targeted VM on same physical machine called as simple side channel attack. This paper introduces how to avert the side channel attack in cloud computing. This is accomplished by using combination of Virtual firewall appliance and randomly encryption decryption (using concept of confusion diffusion) and provide RAS (Reliability, Availability, and Security) of client's data or information.

Keywords- Cloud computing, side channel attack, Amazon EC2 service case study, virtual firewall appliance, randomly encryption decryption.

I. INTRODUCTION

Cloud computing is a word that delivering hosted service over the internet. Cloud computing is the use of computing resource (hardware and software) that are delivered as a service over an internet network.

Cloud computing architecture as show in figure 1 is divided into two sections: Front end and Back end. They connect to each other through network, usually internet. The front end side is computer user or client and back end is cloud provider. The front end includes the client's computer and the application required to access the cloud computing system. On the back end of the system are the various computers, virtual machines (VMs), servers and data storage system that create the cloud of computing service.

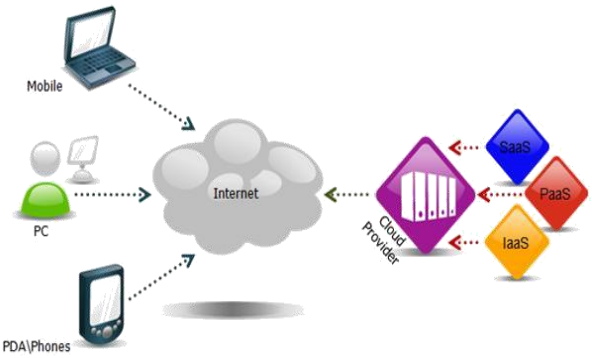

Figure 1: Cloud computing architecture

Manuscript Received on December, 2012.

Bhrugu Saevak, M.Tech. Scholar, Department of Information Technology, Parul Instute of Engineering and Technology, Vadodara(Gujrat), India

Cloud computing is deployed as three model such as Public Cloud, Private Cloud, Hybrid Cloud. Public Cloud: A public cloud is one based on the standard cloud computing model in which a service provider makes resources such as application and storage available to general public over internet. Public cloud services may be free or offered on pay-per-usage model. Private Cloud: It is also called as internal cloud or corporate cloud. Private cloud is cloud infrastructure operated for single organization and managed by third party and hosted internally or externally. Hybrid Cloud: A hybrid cloud is a composition of at least one private cloud and at least one public cloud (combination of both public and private cloud). It is a cloud computing environment in which an organization provides and manages some resources in house and has others provided externally.

These services are classified into three types: Infrastructure as a service (Iaas), Platform as a service (Paas), and Software as a service (SaaS). Infrastructure as a service (Iaas): This is most basic cloud service model like providers offer computers as physical or more virtual machine and other resources. The virtual machine are run as guests by a hypervisor or Virtual machine Manager or monitor(VMM).Platform as a service (PaaS): In this cloud service model cloud providers delivers a computing platform like operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solution on a cloud platform without the cost of buying and managing the under laying hardware and software. With some PaaS offers, the under laying computer and storage resources scale automatically to match application demand such that cloud user does not have to allocate resources manually. Software as a service (SaaS): In this cloud service model cloud providers install and operate application software in the cloud and cloud users access the software. Some type of cloud based application software like Desktop as a service (Daas), business process as a service, and communication as a service.

II. SIDE CHANNEL ATTACK

Infrastructure as a Service(IaaS) model in cloud computing provides infrastructures like a collection of multiple computers, virtual machines(VMs) and other resources to its users to store their application, file, confidential information, documents and so on. Using the Amazon EC2 service as a case study, it is possible to map the internal cloud infrastructure and identify where a particular target VM is likely to reside, and then instantiate new VMs until one is placed co-resident with the target VM. After the successfully placement of instantiate VM to targeted VM then extract the confidential information from the targeted VM called as a Side channel attack. Side channel attack requires two main steps: Placement and Extraction. Placement refers to the adversary or attacker

arranging to place their malicious VM on the same physical machine. Extraction: After successfully placement of the malicious VM to the targeted VM extract the confidential information, file and documents on the targeted VM. There are numbers of ways for such attack but n this paper I focus on side channel attack.

III. AMAZON'S EC2 SERVICE

Amazon's Elastic Compute Cloud (EC2) service, which enables users to flexibility, rent computational resources for use by their application. EC2 provides the ability to run Linux, FreeBSD, OpenSolaris and Windows as guest operating systems within a virtual machine (VM) provided by a version of the Xen hypervisor. The hypervisor plays the role of a virtual machine monitor and provides isolation between VMs, intermediating access to physical-memory and devices. A privileged virtual machine, called Domain0 (Dom0).

When first registering with EC2, each user creates an account uniquely specified by its contact e-mail address and provides credit card information for billing compute and I/O - charges. With a valid account, a user creates one or more VM images, based on a supplied Xen-compatible kernel, but with an otherwise arbitrary configuration. He can run one or more copies of these images on Amazon's network of machines. One such running image is called an instance, and when the instance is launched, it is assigned to a single physical machine within the EC2 network for its lifetime. By default, each user account is limited to 20 concurrently running instances.

3.1 VM CO-Rsidence And Placement:

Understanding VM placement in the EC2 system and achieving co-resident placement for an adversary. Use of network probing both to identify public services hosted on EC2 and to provide evidence of co-residence so utilize nmap, hping, and wget to perform network probes to determine liveness of EC2 instances. Use of nmap to perform TCP connects probes, which attempt to complete a 3-way hand-shake between a source and target. Use of hping to perform TCP SYN trace routes, which iteratively sends TCP SYN packets with increasing time-to-lives (TTLs) until no ACK is received. Both TCP connect probes and SYN trace routes require a target port; we only targeted ports 80 or 443. Use of wget to retrieve web pages, but capped so that at most 1024 bytes is retrieved from any individual web server. Two types of probes: external probes and internal probes. A probe is external when it originates from a system outside EC2 and has destination an EC2 instance. A probe is internal if it originates from an EC2 instance and has destination another EC2 instance.

Determining CO_RESIDENCE checks by exploiting a hard disk based covert channel between EC2 instances.

3.1.1 Network based co-residence checks:

Using our experience running instances while mapping EC2 and inspecting data collected about them, we identify several potential methods for checking if two instances are co-resident. Namely, instances are likely co-resident if they have

(1) Matching Dom0 IP address,
(2) Small packet round-trip times, or
(3) Numerically close internal IP addresses

As mentioned, an instance's network traffic's first hop is the Dom0 privileged VM. An instance owner can determine its Dom0 IP from the first hop on any route out from the instance. One can determine an uncontrolled instance's Dom0 IP by performing a TCP SYN trace route to it from another instance and inspecting the last hop. For the second test, we noticed that round-trip times (RTTs) required a "warm-up": the first reported RTT in any sequence of probes was almost always an order of magnitude slower than subsequent probes. Thus for this method we perform 10 probes and just discard the first. The third check makes use of the manner in which internal IP addresses appear to be assigned by EC2. The same Dom0 IP will be shared by instances with a contiguous sequence of internal IP addresses.

3.1.2 Veracity of the co-residence checks:

We verify the correctness of our network-based co-residence checks using as ground truth the ability to send messages over a cross-VM covert channel. If two instances can successfully transmit via the covert channel then they are co-resident, otherwise not.

3.1.3 Obfuscating co-residence:

A cloud provider could likely render the network-based co-residence checks we use moot. For example, a provider might have Dom0 not respond in trace routes, might randomly assign internal IP addresses at the time of instance launch, and/or might use virtual LANs to isolate accounts. If such precautions are taken, attackers might need to turn to co-residence checks that do not rarely on network measurement.

In previous section determining the CO-RESIDENCE next step is checking whether VM is placement to the targeted VM on same physical machine. In this section we assess the feasibility of achieving co-residence with such target victims, saying the attacker is successful if he or she achieves good coverage.

Before we describe strategies, we first collect several observations we initially made regarding Amazon's placement algorithms. Subsequent interactions with EC2 only reinforced these observations. A single account was never seen to have two instances simultaneously running on the same physical machine, so running n instances in parallel under a single account results in placement on n separate machines.

3.1.4 Brute-forcing placement:

In brute-forcing placement the attacker enumerates a set of potential target victims. The adversary then infers which of these targets belong to a particular availability zone and is of a particular instance type using the map then, over some period of time the adversary repeatedly runs probe instances in the target zone and of the target type. Each probe instance checks if it is co-resident with any of the targets. If not the instance is quickly terminated.

3.1.5 Abusing placement locality:

This strategy Abusing placement locality is doing better than brute-force placement for individual targets or small target sets. Discuss this strategy we assume that an attacker can launch instances relatively soon after the launch of a target victim. The attacker then engages in instance flooding: running as many instances in parallel as possible in the appropriate availability zone and of the appropriate

type. While an individual account is limited to 20 instances, it is trivial to gain access to more accounts. As we show, running probe instances temporally near the launch of a victim allows the attacker to effectively take advantage of the parallel placement locality exhibited by the EC2 placement algorithms.

But why would we expect that an attacker can launch instances soon after a particular target victim is launched? Here the dynamic nature of cloud computing plays well into the hands of creative adversaries. Recall that one of the main features of cloud computing is to only run servers when needed. This suggests that servers are often run on instances, terminated when not needed, and later run again. So for example, an attacker can monitor a server's state, wait until the instance disappears, and then if it reappears as a new instance, engage in instance flooding. Even more interestingly, an attacker might be able to actively trigger new victim instances due to the use of auto scaling systems. These automatically grow the number of instances used by a service to meet increases in demand. We believe clever adversaries can find many other practical realizations of this attack scenario.

3.1.6 Patching placement vulnerabilities:

The EC2 placement algorithms allow attackers to use relatively simple strategies to achieve co-residence with victims. As discussed earlier, inhibiting cartography or co-residence checking would seem insufficient to stop a dedicated attacker. On the other hand, there is a straightforward way to "patch" all placement vulnerabilities: offload choice to users. Namely, let users request placement of their VMs on machines that can only be populated by VMs from their accounts. In exchange, the users can pay the opportunity cost of leaving some of these machines under-utilized. In an optimal assignment policy this additional overhead should never need to exceed the cost of a single physical machine.

3.2 VM Extraction:

The previous sections have established that an attacker can often place his or her instance on the same physical machine as a target instance. In this section, we show the ability of a malicious instance to utilize side channels to learn information about co-resident instances.

3.2.1 On stealing cryptographic keys:

In this type of attack, in the context of third-party compute clouds, would be incredibly damaging and since the same hardware channels exist, are fundamentally just as feasible. In practice, cryptographic cross-VM attacks turn out to be somewhat more difficult to realize due to factors such as core migration, coarser scheduling algorithms, double indirection of memory addresses. The side channel attacks we report on in the rest of this section are more coarse-grained than those required to extract cryptographic keys. While this means the attacks extract less bits of information, it also means they are more robust and potentially simpler to implement in noisy environments such as EC2.

IV. INHIBITING SIDE-CHANNEL ATTACKS

This paper mainly focus on the defense against the vulnerabilities of side channel attack in cloud computing. This might be accomplished by the combination of firewall and random encryption decryption (using concept confusion and diffusion). As per previous section of side channel attack two steps are required to perform side channel attack. Placement and Extraction. To prevent the side channel attack we must to prevent these two steps, so for preventing first step Placement, we implement the virtual firewall appliance in the backend of the cloud computing and for preventing second step Extraction, we use the randomly encryption decryption.

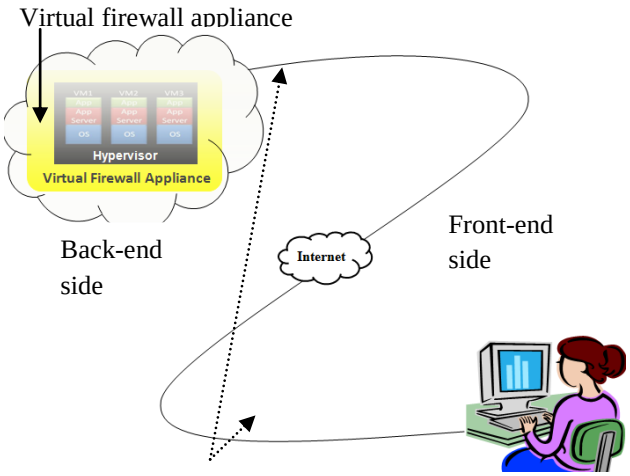

Fig: Security model for side channel attack

Randomly encryption decryption

4.1 Virtual Firewall Appliance:

Firewall is a set of related programs that protects the resources of users from other networks and intruders or adversaries. Here we implement virtual firewall in the cloud server back end of the cloud computing. Now as per Amazon EC2 service case study it is possible to adversaries or intruders identify the targeted VM in cloud infrastructure and then instantiate new VM to targeted VM and extract confidential information but we implement virtual firewall in cloud server so when adversaries identify targeted VM in cloud infrastructure and then place an instantiate VM to targeted VM, virtual firewall prevent this placement step in side channel attack because of we implement virtual firewall in cloud server.

4.2 Randomly Encryption Decryption:

After implement virtual firewall appliance adversaries not place VM to targeted VM so we prevent the side channel attack via virtual firewall but now days cloud computing services are already used for e-commerce applications, medical record services, and back-office business applications , all of which require strong security guarantees. For provide more security we use randomly encryption decryption using concept of confusion and diffusion for prevent second step extraction of side channel attack. Confusion refers to making the relationship between the plaintext and the ciphertext as complex and involved as possible; diffusion refers to the property that the redundancy in the statistics of the plaintext is "dissipated" in the statistics of the ciphertext. In other words, the non-uniformity in the distribution of the

individual letters in the plaintext should be redistributed into the non-uniformity in the distribution of much larger structures of the ciphertext, which is much harder to detect. In randomly encryption decryption, front end side of cloud computing architecture, client's confidential information, important file and documents are encrypted by encryption algorithm which using concept of confusion diffusion like Data Encryption Standard (DES), 3DES, Advance Encryption Standard (AES), Feistel encryption.

Randomly encryption decryption means front end side of client's data or information encrypted through different encryption algorithm which used concept of confusion diffusion and as per National Institute Of Standard And Technology (NIST) AES, DES, 3DES are most secure algorithm for encryption decryption. For using randomly encryption decryption each and every time client's data or information encrypted through different encryption algorithm so adversaries or intruders have more difficulties to detect or extract cryptography key and encrypted data sent over internet network to back end side of cloud computing

Using combination of virtual firewall and randomly encryption decryption prevent two step of side channel attack and provide security against side channel attack and provide reliability, scalability, and security (RSA) of data or information.

V. CONCLUSION

Using side-channel attack, it can be very easy to gain secret information from a device so it is good idea to provide security against side channel attack in cloud computing using combination of virtual firewall appliance and randomly encryption decryption (using concept of confusion diffusion) because it provides security against both front end and back end side of cloud computing architecture and also provide RAS (Reliability, Availability, and Security)

REFERENCES

[1] http://en.wikipedia.org/wiki/Cloud_computing
[2] http://searchcloudcomputing.techtarget.co/ Security Analysis of Cloud Computing
[3] Brodkin, J.: Seven Cloud Computing Security Risks(2008) http://www.gartner.com/DisplayDocument?id=685308
[4] http://cloudsecurity.org/
[5] Hey, You, Get Off of My Cloud - Computer Science and Engineering cseweb.ucsd.edu/~hovav/dist/cloudsec.pdf
[6] bAmazon Elastic Compute Cloud (EC2). http://aws.amazon.com/ec2/
[7] Amazon Web Services. Customer Agreement. http://aws.amazon.com/agreement/
[8] Virtual firewall - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Virtual_firewall
[9] Virtual Firewall Appliances: Trust Misplaced? Cloud Passage Blog blog.cloudpassage.com/.../virtual-firewall-appliances-trust-misplaced/
[10] Cloud Security Alliance Guidance, "Security Guidance For Critical Areas of Focus In Cloud Computing Vl.0", www.cloudsecurityalliance.orglguidance/csaguide.v1.0.pdf, published April 2009
[11] National Institute of Science and Technology. "The NIST Definition of [15] Luis M. Vaquero1, Luis Rodero-Merino1, Juan Caceres1, Maik Cloud Computing".p.7. Retrieved July 24 2011.
[12] Shannon's Idea of Confusion and Diffusion www.cs.ust.hk/faculty/cding/COMP581/SLIDES/confdiffu.pdf