Cloud images security problems and proposed solutions

Mohammad-Reza Memarian

24 September 2014

1 Introduction

Virtual machine (VM) images construct fundamental of security in cloud computing. Instances are deployed using either paid or shared VM images. As a result, integrity and security of VM images and secure management of image repositories are factors that affect security of cloud computing. When a customer aims to hire Infrastructure as a service (IaaS) from a provider, he should hire atleast an instance. Then an image is required to be run on the instance. He can either import his own image from outside of the cloud or he should use from existing images in the cloud. Existing images in the cloud are either shared or paid VM images. Shared VM images are the ones which are published and shared by other users in the cloud. Paid images are images that are created by third-party developers and is on sale to other interested users. Image publishers can either be normal third-party users or official creditable entities.

In general, using either shared or paid images would have risky consequences for image retriever. There are two problem statement thought and discussed by the author which are about security of VM images used in cloud computing. Those two problem statements are discussed in the following sections.

2 Detection of Malicious VM images and respective publisher using VM introspection and anomaly detection techniques

As any image publisher share his image wither to a limited number of users or public, a malicious image publisher also shares his malicious image to the public. His image will be used by several users that their instances can be distributed over various cloud clusters. Indeed if malicious images is interesting enough to be used by several users, image owner will have several zombies running on different instances which are located in different locations in the cloud as shown in Figure 1. Based on Amazon website [1] (An IaaS provider), there is no easy way to know who has provided a shared image, because each image is

represented by an account ID. So Amazon encourages Image publishers to give a description of their images in related Amazon forum.

So In Figure 1 four steps shown can be interpreted as below:
Step 1, Malicious user requests through image API located in image component to share his image with public.
Step 2, Malicious image is stored in object storage (Repository).
Step 3, Alice and Anna from Image component to retrieve malicious image for their instances.
Step 4, Instances of Anna and Alice get located in the cloud with having malicious image running on them.

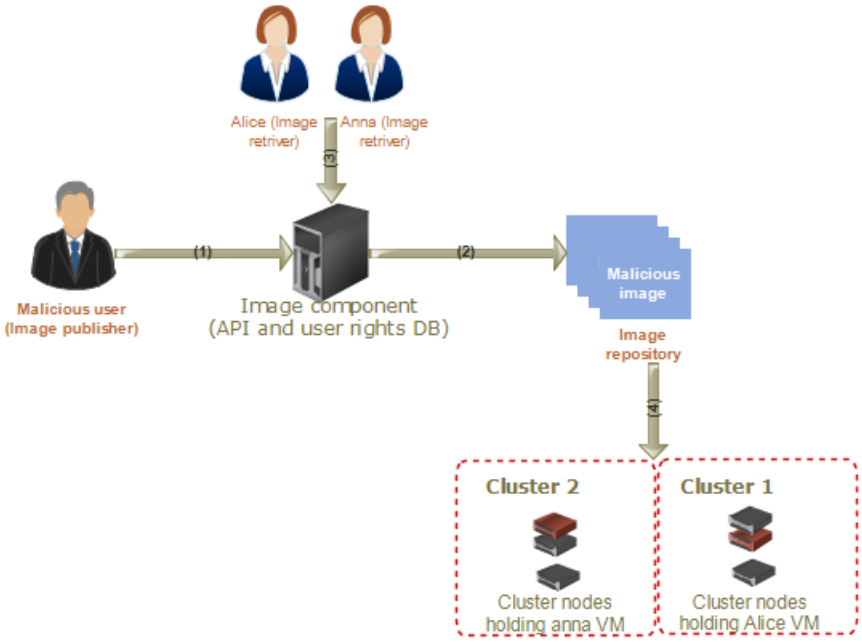

Figure 1: First problem scenario

Different users may use differently from each image in term of computing, as a result each VM may have different performance. So by analyzing behavior of only one of the images published by a publisher, we can not determine whether the publisher is malicious or not.

As shown in figure 2, our aim will be to analyze all the instances which are using image published by the same publisher. By analyzing behavior of all the images, we will construct a table made up of some parameters which represents the general behavior of all instances using a shared image. Virtual Machine Introspection is a technique which allows querying running VMs about theirs status from a trusted source. By using this technique we can gather useful

information to construct a system state table. by statically analyzing all the entries and giving weights to each parameter, either one of the entries which is dominant will be inserted into the table or average of all will be inserted. Using anomaly detection technique of machine learning, the constructed table will then be compared to a normal behavior tables which consist of various combinations of that type. Using this plan, there is possibility that a malicious VM will be detected. The difference of this solution and similar solutions like Intrusion detection system (IDS) is that IDS tries to detect anomalies happening in a node or a cluster or a network as a whole, But our work tries to detect adequate number of malicious VMs at one time based on their common criteria which is same publisher. In figure 2, proposed solution has been sketched to show a better picture of the solution.

So In figure 2, three steps shown can be interpreted as below:
Step 1, Each node's system info's are queered using VM introspection and they are put in a table.
Step 2, Based on the weights which are given to each entity, either the highest one or average of all is placed in unified image state.
Step 3, Unified image state is compared to several standard behaviors to check whether it matches any of those or not.

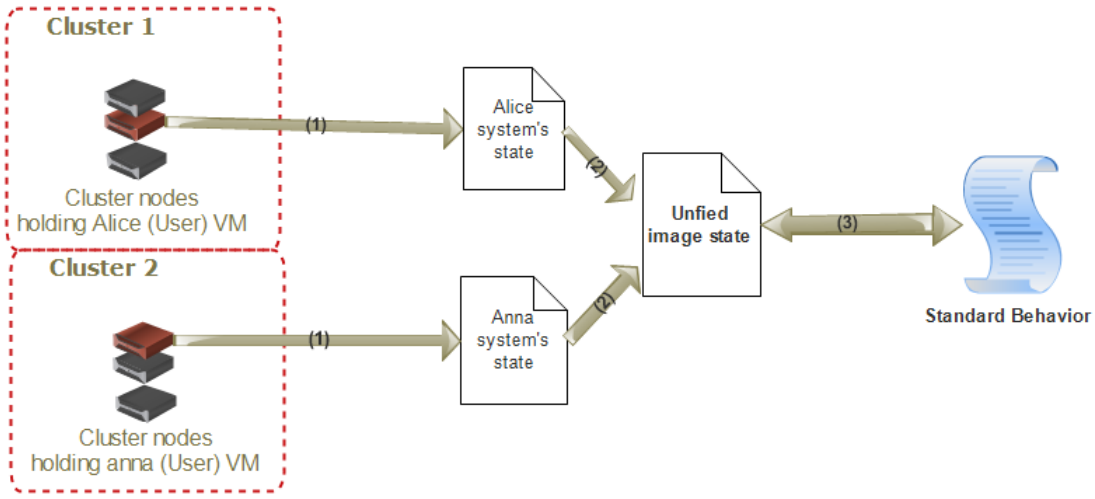

Figure 2: Proposed solution to problem in Figure 1

3 A mechanism to prevent Image robbery from cloud

The images may include confidential information, licenses and critical software stack. As mentioned before, users can share images either publicly or among several intended target user groups. An example of this scenario can be when a company requires all the developers working with it to work with a single standard VM image. A sample work flow for this case is the way rackspace (an IaaS provider) documented the process it uses at [2]. In figure 3, a sample of permission granting process to other users is shown in first three steps. As depicted in figure 3, First three stages can be interpreted as follows:
Step 1, A known user should be listed in image X member list. In the image member list, User IDs will be added to image member list of the image (Image member can be later used for retrieving detail of users in which the image is shared with them).
Step 2, The member in which the image is shared with him must accept that the shared image be placed in his image list.
Step 3, Bob accepts the shared image and is able to use from Alice's image to run his instance.

The problem in this stage is that Bob, may can re-share or re-bundle the image and store it under it's ownership which can be counted as violation of privacy. So proceeding with steps in figure 3, remaining step is as below:
Step 4, Re-shared/re-bundled image which has be done by Bob, either is stored under his ownership in object storage or export the image. It should be mentioned that cloud providers like Amazon and rackspace have stated in their website that users can not export images that has not been imported by them.

Several solutions can be proposed to this problem. In the model shown in figure 3, only User IDs are required to create the image member list. So as an instance, the image owner states in the image member list that the use A with ID 123456 and user B with ID 56789 may have access to his image X. One solution can be to assign an ID to all the instances in the cloud, so each image apart from getting mapped to users IDs, will be mapped to the instances ID as well. This information will be attached to the image in time of export or share. As a result even if the user will be successful to somehow export or re-share the image, he will not be able to run the image on the other instances or other users also can not run this image on any other machine.

The proposed solution is fairly simple which can be effective too. But indeed another component can be added to the solution. Sensitive parts of operating systems can be detected and using cryptography they can be encrypted. Key management and handling will be responsibility of the cloud service provider. So when a user wants to run an image, if the user and instance ID are matched with Image ID, the key is passed to the user to decrypt the encrypted modules. As a result the image can be used only in the predetermined environments. So as depicted in figure 4, when Bob steals the image, it can not be run on any

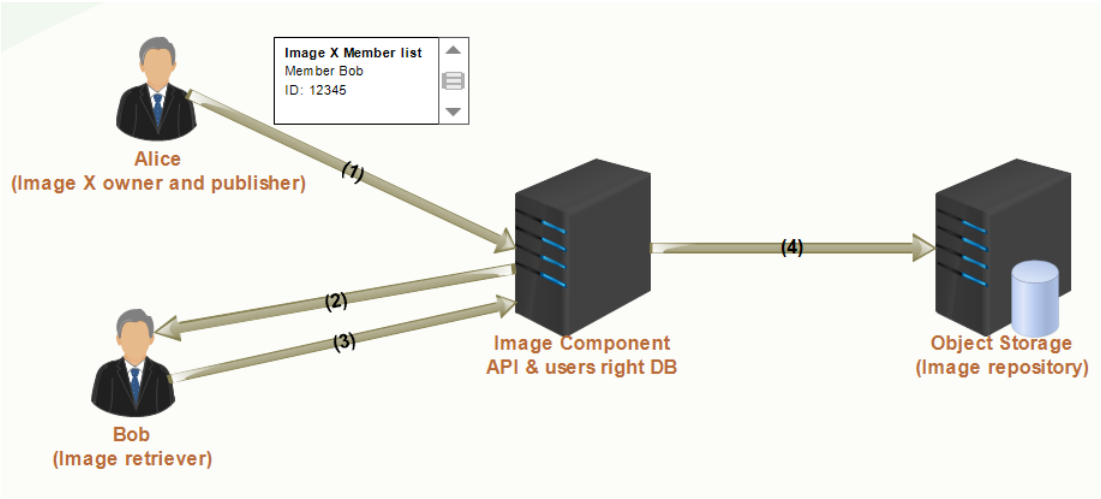

Figure 3: Stealing third-party image

instance except the one that is mentioned in image member list.

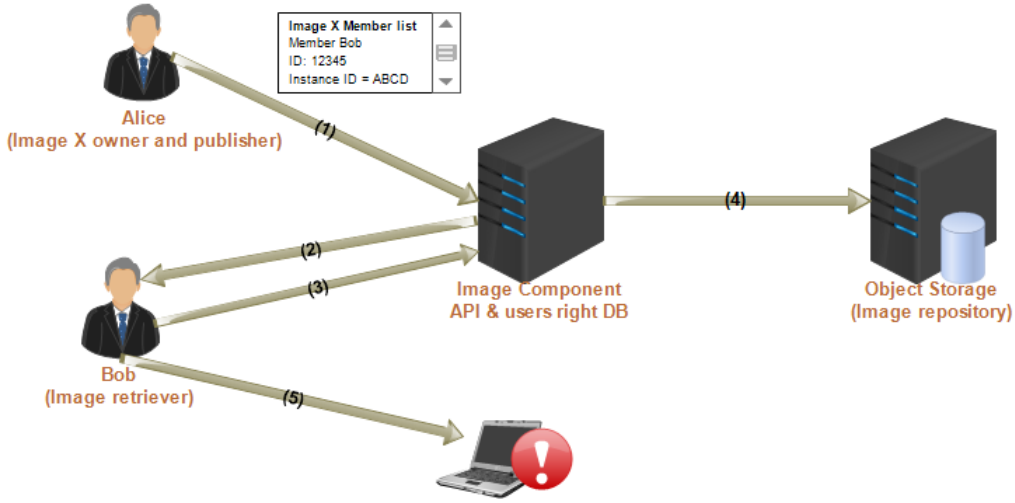

Figure 4: Proposed solution to problem in figure 3

References

[1] *aws.amazon.com/ec2/.*

[2] *http://docs.rackspace.com/images/api/v2/ci-gettingstarted/content/image-sharing.htmlsharing-flow.*