A Demonstration of Malicious Insider Attacks inside Cloud IaaS Vendor

Minh-Duong Nguyen, Ngoc-Tu Chau, Seungwook Jung, and Souhwan Jung

***Abstract*—Until now, many researches have carried out analyzing the vulnerabilities as well as finding the defense strategies for malicious insider (MI) at cloud environment. However, all these previous works only considered the perspective of MI attacks that are originated from tenant side in a public cloud. Furthermore, in these existing works, the MI attack techniques are only basically and abstractly described. Without the proof of concept, MI attacks are just theoretical threats. In this paper, we consider the scenario that MI executes the attack inside the Cloud IaaS vendor. Moreover, in order to show the realistic of MI attacks in the scenario, this paper introduces three concrete MI attacks with a proof of concept implementation based on existing tools. Three introduced MI attacks in this paper are: memory scanning, template poisoning, and snapshot cracking. The demonstration result shows that MI attacks inside cloud IaaS vendor are no longer potential threats but realistic issues that we need to consider.**

***Index Terms*—CloudStack, malicious insider, insider threats, cloud computing, cloud security, security threats.**

I. Introduction

According to ESG Insider Threats Survey 2013, more than half of all survey organizations are extremely vulnerable or somewhat vulnerable to 66% of potential insider attack methods [1]. Moreover, as US State of Cyber Security Survey in 2013 [2], 53% of respondents admitted that damages, which are caused by the insider attacks, are more than outsider attacks. Likewise, Cloud Security Alliance (CSA) carried out researches for cloud computer top threats in 2013. In the report, malicious insider (MI) was mentioned as one of the greatest risk [3]. The famous Edward Snowden incident can be seen as one of the most highlighted topic over the internet security in 2013 that raised the fact that organization's information can be revealed at any point from MIs with bad intention. Another highlighted incident was occurred at Twitter when several companies and personal documents uncovered by Twitter administrator's account that was hacked by MI [4].

Along with reports and researches about the MI threat, many researchers are finding penetration methods to the cloud as well as the way to prevent information leakage and mitigate damages caused by MIs. In [5], authors demonstrated the possibilities of compromising target's data through IaaS Virutal Machine Cloning, DaaS File Copy, and Acid Clouds and Fraud as a Service (FaaS). In the other hand, authors in [6] focused on MI threats to relational databases in cloud. Moreover, in [7], in order to detect malicious usage patterns of MIs, authors have proposed an Insider Threat Detection Model. The above papers are representatives among many research papers, which provided protection and prevention models at tenant side, based on theoretical MI attacks in the cloud. However, these research papers only focus on analyzing the basic and conceptual MI attacks that are not concrete illustration of the attack methods and these are just theoretical attacks.

Although the MI threats on Cloud Service Provider (CSP) are uncertain, tenants have to trust CSP and IaaS security. The question here is: "If there is an MI hiding inside cloud IaaS vendor, is it possible for him to obtain your information or not?"

In order to answer the question, this paper provides three MI attack methods inside cloud IaaS vendor with detailed description of how to conduct attacks as a demonstration. The demonstration will be conducted under CloudStack-based environment, which is one of the most popular open source cloud computing software.

This paper is organized as follows. Section II provides the background information for snapshot mechanism and Linux memory management on CloudStack, which are the main targets for MI attack methods. Section III introduces the concept of MI attack methods. The next section illustrates the proof of concept for MI attack methods. Final section shows the conclusion of this paper.

II. Background

A. CloudStack Architecture

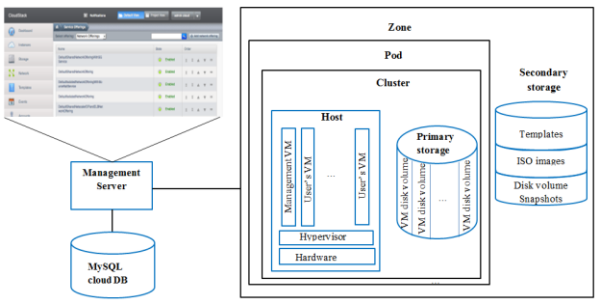

Fig. 1. Conceptual view of CloudStack architecture.

CloudStack, which has been known as one of the most popular open source cloud computing software, was chosen by many cloud vendors for creating, managing, and deploying their infrastructure cloud services. Cloudstack architecture is comprised of Management Server, Hypervisor host, and virtualization network (Fig. 1) [8]. Management

Manuscript received November 5, 2013; revised January 14, 2014.

Minh-Duong Nguyen, Ngoc-Tu Chau, Seungwook Jung, and Souhwan Jung are with Soongsil University, Seoul, Korea, Republic of (e-mail: nguyenminhduong@ssu.ac.kr, chaungoctu@ssu.ac.kr, seungwookj@ssu.ac.kr, souhwanj@ssu.ac.kr).

Server manages cloud resources and stores resources information into cloud database. Hypervisor hosts are managed inside zones, which provide virtual hardware and virtual machines (VM). Data storages are comprised of primary storages and secondary storages, which are used to store the VM disk volume as well as disk image (templates, snapshots, iso files).

B. Snapshot Mechanism on CloudStack

For Cloudstack, VM snapshot and volume snapshot are two different mechanisms. Volume snapshot is similar to disk backup for Cloudstack. In the other side, VM snapshot is a copy of the virtual machine at a given point in time. The requests of snapshot are processed using VMware native facility that follows the below steps:

1) Using Cloudstack API to perform the requests such as creating, removing, or reverting snapshot for a VM from clients to the cloud server.
2) The request is forwarded to a host that includes snapshot-needed VM.
3) The host writes data and state of that VM to disk in qcow2 format and stores it in secondary storage.

Since requests for snapshots in CloudStack are based on VMware native mechanism, it is possible to convert qcow2 format back to vmdk format that is deployable on VMware workstation.

C. Cleartext Passwords in Linux Memory

Cloud servers are mainframes where a lot of users launched sessions and processes are run on the same physical machine. A central piece of the operating system is the kernel which runs at the highest privilege level (regarding to virtualization) and manages privilege levels [9]. VMs run at a lower level and are forcibly prevented by the kernel from reading or writing each other's memory. VMs obtain RAM by pages from the kernel. Memory page sharing is supported though a kernel feature–Kernel Same Page Merging (KSM). KSM handles memory pages and merges the memory pages of a VM into a single page and shares it to other VMs. When running many VMs on a host, memory pages are shared. Some critical data of each VM is purportedly held in memory and made it to a physical memory where it will stay until overwritten. Because of that, it is possible for MI to inspect the data. From cold boot attack, they demonstrated that Linux memory stores plain text passwords including login, SSH, Email, Truecrypt and root passwords. And the passwords can be recovered from memory in plaintext [10]. From this achievement, attacker can take attacking advantage to target user.

III. Malicious Attacks in Cloud Vendor

In this section, some attack methods are presented in detail in order to explain how MI can get confidential information of tenants.

These attacks assume the following scenarios. In the 1st attack method, the attacker is assumed as a malicious insider that has root access to hypervisor host server. In the 2nd and 3rd methods, the attacker is assumed to have the privilege to access to Storage server. The attack methods are presented with the first attack methods based on memory dump [11] and the next attack methods based on VM snapshot attack.

A. Memory Dump Scanning

As mentioned in background, memory dump shows the clear text including login, root password, mail, SSH information, or even private keys information. In this attack, attacker can easily get some information from dumping memory. When retrieving information in a large amount of data, critical data is hidden in at least hundreds of megabytes of data. Actually, it is difficult for MI to find the confidential data without keywords for scanning. In order to increase success rate, MI also can use social engineering techniques, refers to psychological manipulation between MI and target organizations such as email, contract, public information of target organization. From that, he can collect necessary information. After that, by applying the social engineering information or dictionary attack, MI can guess the confidential information such as root password, user login…

B. Templates Poisoning

Cloud IaaS vendor usually provides default public template that is the prepared common operating system. Employees who have access privilege to storage server or cloud management web interface (Web UI) can download the default public templates. Template poisoning attack assumes the scenario when MIs, who have enough privileges to access the storage server or Cloud management Web UI, download the template and deploy the downloaded template in his private server with the attempt to poison the template. The poisoned template will be uploaded back into the Cloud. After MIs successfully uploaded the poison template, virtual machines that are deployed from the poisoned template are vulnerable to MIs.

C. Snapshot Cracking

There are many cloud vendors with different business models and architectures such as Amazon EC2, Microsoft Azure, Google Cloud, KT Ucloud... But not all cloud vendors encrypted their VM disks. VM disk encryption also meets some limitations such as shared resources with other tenants, mounting data [12]. Many cloud vendors still manage their VMs which only use normal user and password. If MI is a VM administrator or system administrator who can take a snapshot of target VM, he can easily make an attack on snapshot of that VM.

IV. Proof of Concept

The environment of our implementation was Cloudstack architecture with KVM hypervisor server that is used as host computer. Cloudstack was installed on a centos 64-bit server and worked as Management Server. Another centos 64-bit server was installed and acted as both primary and secondary storages. Host's specification was an Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz and 16G RAM.

A. Memory Dump Scanning

To perform the 1st attack, we used *virsh* and *dump* command to perform dump memory region that contains the target VM's information. With a prepared dictionary, we

used *strings* and *grep* commands to find specific information. This attack is performed in Fig. 2.

```
#virsh
#dump target-vm-name targetvm.dump
#exit
#cat >> getpwd.sh << EOF
 #!bin/bash
 @dictionary=
{loginpwd1,loginpwd2,…,loginpwdn};
 foreach pwd in @dictionary{
 less targetvm.dump|strings|grep pwd >pwd.txt
 }
EOF
#chmod +x getpwd.sh
#./getpwd.sh
```

Fig. 2. Attacker finds clear text in memory dump of target VM.

B. Public Templates Poisoning

To perform the 2nd attack, we used download template function on Cloudstack UI. For alternative method, we can obtain the template by connecting directly to the storage server. All public templates are stored in secondary storage. In order to find the downloaded template file, we go to the storage server and find the mounted partition for secondary storage by using the *exportfs* command. All template folders contain the "*template.properties"* file which is acted as a descriptor for the template file. Using the simple linux command *grep*, we can extract the information from "template.properties" file.

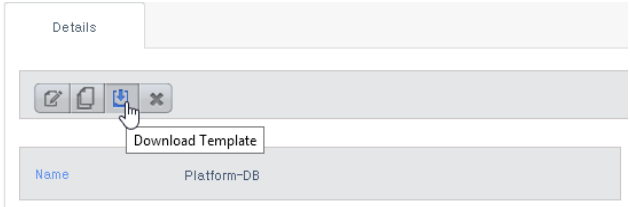

Fig. 3. Template can be downloaded from CloudStack UI, or copied directly from secondary storage.

```
[root@s4urc-sto 249]# grep -r "CentOS 5.5" /export/
/export/secondary/template/tmpl/1/4/template.properties:description=CentOS 5.5(6
4-bit) no GUI (KVM)
```

Fig. 4. Finding public templates based on the description.

For the next step, we deploy the template in a new virtual machine, create a backdoor daemons for sniffing packet information or tenant's password for specific application, export as template again, and copy back into the secondary storage.

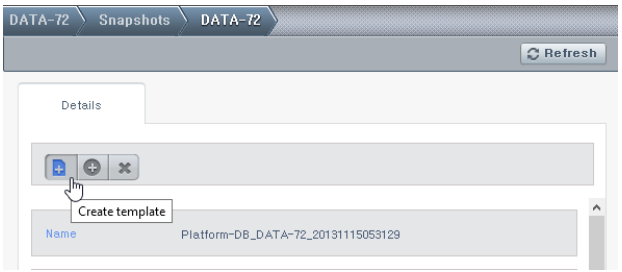

Fig. 5. A template can be created from a snapshot.

C. Snapshot Cracking

In our test-bed, VM snapshot was not encrypted. To perform this attack, at the beginning we took the snapshot of target VM and copied it to another server. Because test-bed used KVM hypervisor host, snapshot is stored in qcow2 format. The second step, we converted the snapshot from qcow2 to vmdk. Vmdk is a file format of Vmware.

For the third step, we used Vmware workstation to boot the converted snapshot with a bootable tool (System Rescue Cd)–a tool for administrating or repairing the system or data after a crash or recovering root password.

```
[root@s4urc-sto 201]# ls
15b8e8ca-a76b-3f47-971e-b397f01e4df3.qcow2  template.properties
[root@s4urc-sto 201]# qemu-img convert 15b8e8ca-a76b-3f47-971e-b397f01e4df3.qcow2
 -O vmdk targetVM.vmdk
```

Fig. 6. Target VM's snapshot and converting from qcow2 format to vmdk format.

```
root@sysresccd /root % fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0009310a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    18874367     9436160   83  Linux
/dev/sda2        18876414    20969471     1046529    5  Extended
/dev/sda5        18876416    20969471     1046528   82  Linux swap / Solaris
root@sysresccd /root % mount /dev/sda2 /mnt/_
device or mount point
sda#   sda1#  sda2#  sda5#
```

Fig. 7. Using system rescue Cd to show and mount the disk partions of target VM.

For the final step, after booting that snapshot with System Rescue Cd and mounting snapshot disk into system like Fig. 4, we broke password of that snapshot by changing the root password in shadow file, as shown in Fig. 8. After that, we got all information in that snapshot of target VM.

```
root:!:13479:0:99999:7:::
daemon:*:15506:0:99999:7:::
bin:*:15506:0:99999:7:::
sys:*:15506:0:99999:7:::
sync:*:15506:0:99999:7:::
games:*:15506:0:99999:7:::
man:*:15506:0:99999:7:::
lp:*:15506:0:99999:7:::
mail:*:15506:0:99999:7:::
```

Fig. 8. Changing root password in/ mnt/etc/shadow file.

V. Conclusion

In these existing works, the MI attack techniques are only basically and abstractly described. Without the proof of concept, MI attacks are just theoretical threats. This paper provided the proof of concepts for three MI attacks including: memory scanning, template poisoning, and snapshot cracking. The demonstration results show that MI attacks inside CSP are no longer theoretical threats but these are realistic issues.

Acknowledgment

This research was supported by the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the ITRC(Information Technology Research Center)) support program (NIPA-2013-H0301-13-1003) supervised by the NIPA(National IT Industry Promotion Agency).

This research was supported by the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the ITRC(Information Technology Research Center)) support program (NIPA-2013-H0301-13-1003) supervised by the NIPA(National IT Industry Promotion Agency)

This work was supported by the IT R&D program of ACT under the MOTIE/KEIT. [10045904, The development of Fundamental Technology for Security as a Service (SecaaS) Framework under cloud computing environment and the implementation of 1Gbps mobile data loss prevention (DLP) service based on the SecaaS Framework.]

This work was supported by the IT R&D program of MOTIE/KEIT. [10047320, Developing disaster recovery solution for Disaster Recovery as a Service(DRaaS) in Cloud-Computing, Environment]

REFERENCES

[1] J. Oltsik, *Vormetric/ ESG Insider Threats Survey*, Sep. 2013.
[2] CERT Insider Threat Center, "How bad is the insider threat?," *US State of Cybercrime Survey*, 2013.
[3] CSA, "The notorious nine," *Cloud Computing Top Threats in 2013*, Feb 2013.
[4] M. Arrington. (July 2009). In our box: Hundreds of confidential twitter documents. [Online]. Available: http://techcrunch.com/2009/07/14/in-our-inbox-hundreds-of-confidential-twitter-documents/
[5] A. J. Duncan, S. Creese, and M. Goldsmith, "Insider attacks in cloud computing," in *Proc. 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 25-27 June 2012, pp. 857-862.
[6] Q. Yaseen and B. Panda, "Tackling insider threat in cloud relational databases," in *Proc. 2012 IEEE Fifth International Conference on Utility and Cloud Computing (UCC)*, 5-8 Nov. 2012, pp. 215-218.
[7] N. Lucky, T. Paul, and A. O. Matthew, "Insider threat detection model for the cloud," *Information Security for South Africa*, pp. 1, 8, 14-16 Aug. 2013.
[8] Apache Cloudstack 4.2.0–CloudStack Administrator's Guide. [Online]. Available: https://cloudstack.apache.org/docs/en-US/Apache_CloudStack/4.2.0/html/Admin_Guide/index.html.
[9] Red Hat. KVM–KERNEL BASED VIRTUAL MACHINE. [Online]. Available: http://www.redhat.com/rhecm/rest-rhecm/jcr/repository/collaboration/jcr:system/jcr:versionStorage/5e7884ed7f00000102c317385572f1b1/1/jcr:frozenNode/rh:pdfFile.pdf.
[10] S. Davidoff, *Cleartext passwords in Linux memory*, Massachusetts Institute of Technology, July 2008.
[11] F. Rocha and M. Correia, "Lucy in the sky without diamonds: Stealing confidential data in the cloud," in *Proc. 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 27-30 June 2011, pp. 129-134.
[12] M. Bamiah, S. Brohi, S. Chuprat, and M. N. Brohi, "Cloud implementation security challenges," in *Proc. 2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM)*, 8-10 Dec. 2012, pp. 174-178.

Minh-Duong Nguyen received the BEng degree in electronics and telecommunications from the Ha Noi University of Science and Technology in 2012. Since 2013, he has joined Communication Network Security Laboratory and applied for master course in Information and Telecommunication Engineering at Soongsil University, Seoul, Korea.

He has one year experience as a research assistant about SDN in ECODANE project financed by International Bureau of the BMBF. His current research interests include Cloud computing, Cloud security, Security, Software Defined Network.

Ngoc-Tu Chau received the Msc degree in information and telecommunications from Soongsil University in 2012. Since 2013, he has joined Communication Network Security Laboratory and applied for PhD course in Computer automation and Network at Soongsil University.

Before going to Soongsil University, he worked for Toshiba Software Development in Vietnam co.ltd for 3 years as senior software engineer (2 years) as well as team leader (1 year) and at the same time, as network administrator position (1 year). His current researches include Cloud Computing, Cloud Security as well as SecaaS.

Seungwook Jung received the the PhD degree in engineering from the University of Siegen of German in 2006, is currently a senior research associate at Korea Internet and Security Agency(KISA), and is currently a adjunct professor at the Soongsil Unversity of Korea. His research interests include the information security and privacy in cloud computing and in smart device such as cryptography, information security system architecture, identity management, and the development of national-wide privacy policy.

Souhwan Jung received the B.S. and M.S. degrees in electronics engineering from Seoul National University in 1985 and 1987, respectively, and the Ph.D. degree from the University of Washington, Seattle, USA in 1996. From 1996 to 1997 he was a senior software engineer at Stellar One Corporation, Bellevue, USA. In1997, he joined the School of Electronic Engineering at Soongsil University, Seoul, Korea, and currently serves as a professor.

He is an executive director of the Korea Institute of Information Security and Cryptology. He was also a R&D program director of Ministry of Knowledge Economy in Korea for information security area from 2009 to 2011. His research area includes vehicular network security, VoIP security, wireless network security, and RFID security.