Defense Technique against Spoofing Attacks using Reliable ARP Table in Cloud Computing Environment

Hyo Sung Kang, Jae Hyeok Son, Choong Seon Hong
Networking Lab, Department of Computer Engineering
Kyung Hee University, Korea
{ kanghs, sonjaehyeok, cshong }@khu.ac.kr

***Abstract*— Recently cloud service has been introduced in order for many enterprises to achieve purposes such as improvement in efficiency, cost reduction and revolution in business process. However spoofing or poison attacks on VM inside the cloud cause the deterioration of cloud system and those attacks can make the huddle for spreading the cloud services. Many researches are now under way to solve such problems but most of these seem to be passive and limited in terms of detecting attacks and applying to large scale of networks. In this paper, we propose a defense technique for loss of VM resources against the network attacks called spoofing of poison on OpenStack environment. In our proposal, we use reliable ARP table which makes our proposal more reliable in cloud computing environment. Finally we conclude this paper showing that the proposed mechanism is an effective way to defend the ARP spoofing attack**

Keywords—ARP, Spoofing, Cloud, OpenStack, Security

I. INTRODUCTION

Many enterprises all around the world use cloud computing service for cost reduction and revolution in work processes. Gartner, IT consulting provider, mentioned cloud computing technique among the top 10 strategic technology trends for 2015[1]. This implies that cloud computing technique becomes the core technology in IT field but many companies or institutional users hesitate to utilize cloud service due to the concern about the security.

No.	Threatening Element
1	Data Breaches
2	Data Loss
3	Account or Service Traffic Hijacking
4	Insecure Interface and APIs
5	Denial of Service
6	Malicious Insiders
7	Abuse of Cloud Service
8	Insufficient Due Diligence
9	Shared Technology Vulnerability

Table 1. The Notorious Nine - Cloud Computing Top Threats

There are 9 top threats to cloud computing introduced in Cloud Security Alliance(CSA) in Table 1[2]. Among these, Data Breaches, Data Loss and Account or Service Traffic Hijacking can be the main cause of decrease in the performance of entire cloud system, considering the characteristics of cloud system. Moreover, these threats can be easily realized in the form of external network attacks such as Spoofing or Poison. As a result, cloud security has got great attention by the research community [3] but protocols proposed in the existing work seem to be impractical [4]. The other problem is that it is not scalable to large size of networks [6]. In this paper, we build up cloud computing environment using open source cloud platform called 'OpenStack'. On the top of it, we propose a defense technique against Spoofing attacks using reliable ARP table.

The rest of this paper is made up of the following: Section 2 covers the background of Spoofing attacks. In section 3, we describe the related works. Section 4 deals with the proposed defense technique. Afterwards, section 5 covers the performance evaluation and results. Finally, the conclusion of the work of this paper appears in section 6.

II. ARP AND ARP SPOOFING IN CLOUD ENVIRONMENT

A. Address Resolution Protocol (ARP)

ARP is the standard protocol that converts the addresses of network layer to the addresses of data link layer. It is used to find MAC address when there is no corresponding MAC address for IP address in ARP table. Fig. 1 shows the update process of ARP table. First of all, Host(1), the source node, looks up the table to check the MAC address of the destination node Host(4). If there is no cached address, Host(1) broadcasts ARP Request Message to sub-networks which Host(1) is belonging to. Host(4) receives the request message and replies with ARP Reply Message to Host(1) by unicating. On the other hand, Host(2) and Host(3) drops the request messages since the destination IP addresses are not matched. After Host(1) receives the reply, it updates the ARP table and sends data frames to Host(4) using the MAC address of Host(4).

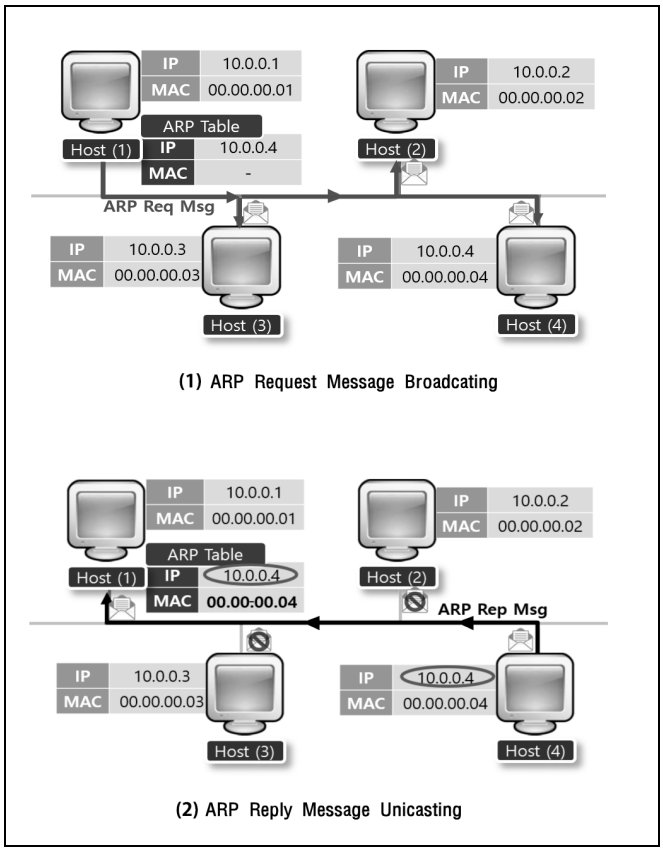

Fig 1. ARP Table Updating Process

As we can see in Fig. 1 there is no extra authentication mechanism in ARP process. However this may lead to serious problems in terms of security. For example, all nodes in the same network can see the ARP Request Message, so it is possible that nodes that should not reply can send Reply Message. In this case, there is no method to block or verify such replies.

B. ARP Spoofing Attack

One of the typical network attack using ARP is ARP Spoofing. It attacks victims by sending modified ARP Reply Message so that the victims misrecognize MAC address of a certain host. This attack disturbs normal communication between hosts. Fig. 2 shows the process of ARP Spoofing. In order to communicate with Host(2), Host(1) needs to know the MAC address of Host(2). Likewise Host(2) should know the address of Host(1) for their communication. As it is mentioned in the previous clause, Host(1) broadcast ARP Request Message to its subnet. At this time, the attacker on the same network receives the request messages from the two hosts and sends the modified replies which contain MAC address of the attacker to them. This leads to modification of ARP table. As a result, Host(1) recognizes the attacker as Host(2) and Host(2) recognizes the attacker as Host(1). This means that each host believes that they are communicating with each other but the fact is that they are exposed to the attacker and all the information of them is captured by the attacker.

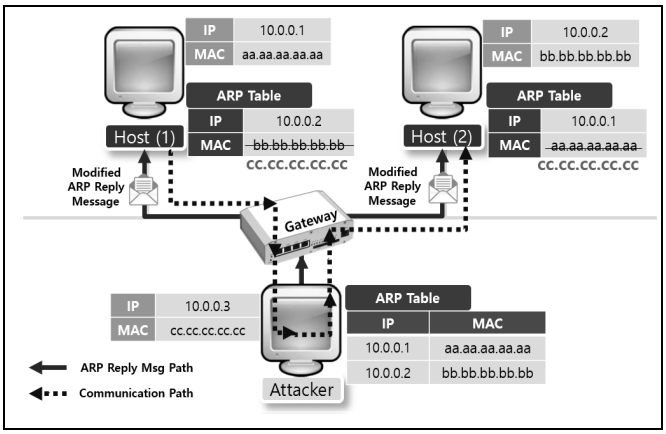

Fig 2. ARP Spoofing Attack

In the cloud environment, the following problems can be caused by ARP Spoofing attack.

- Data Breaches
- Interference of Communication between VMs
- Data Modification
- Loss of Cloud Resources
- Deterioration of Entire Cloud Service

As ARP Spoofing can only be done in the same network as the victims, no special action is required to build the network. However it is essential to have a mechanism to protect the system from Spoofing in cloud environment since hosting service can be provided to many users in a single internal network.

III. RELATED WORK

A. Secure Address Resolution Protocol

What makes ARP Spoofing possible is reception process of ARP Reply Message which does not involve any authentication. As to cope with the problem, Secure Address Resolution Protocol(SARP) was proposed[4]. It adds authentication process to the original ARP. Basically SARP can solve the problem but it is difficult to substitute the existing protocol into completely new protocol in terms of reality.

B. Anti Spoofing Mechanism using OpenFlow

OpenFlow is the protocol that provides the fight to access control Forwarding Table of either switches or routers remotely [5]. In [6], it suggests a solution for ARP Spoofing using OpenFlow. It follows that all the IP address and MAC address information are registered in the controller and ARP Request Message is received by the controller first to compare it with the one that is already present in the controller. This helps to identify modified messages and drops them [6]. These days OpenFlow is used to build cloud network due to its great manageability but it also has problems with respect to

scalability since there are still cloud computing environments to which OpenFlow can not be applied.

IV. PROPOSAL

In this paper, we propose a defense technique that collects IP and MAC addresses of the instances created in OpenStack cloud computing environment, so that reliable ARP tables can be constructed. There are two main parts in our proposed technique, one for creation and management of the ARP tables and the other for ARP Reply Message handling called Comparison Handler. The collected IP and MAC address information is used for the construction of reliable ARP table using authentication service provided in the OpenStack project and stored in the controller node. Furthermore, the message is verified at the controller node. The verification process is the comparison of the reply message with the reliable ARP table. Once there is an unmatched address the packet is regarded as modified so that Spoofing can be detected. So the system can be protected.

A. IP and MAC Information Collection in OpenStack

OpenStack is an open source cloud platform that proposes two ways of installation according to the independence of network service [7]. In this paper, we propose a defense technique in TCP/IP-based cloud environment so OpenStack using Nova-Network is considered to be the best for our proposal.

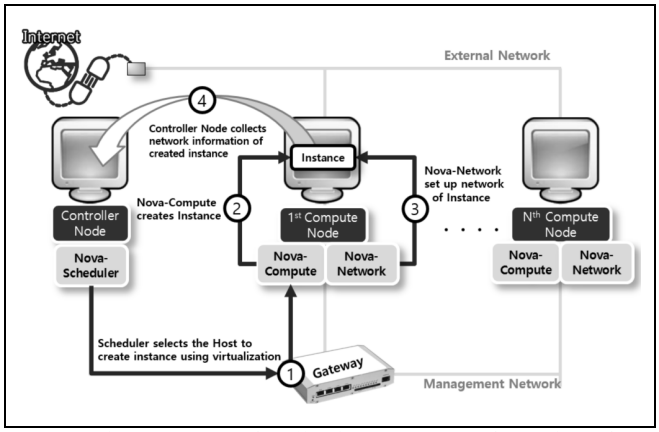

Fig 3. Process of Collecting Network Information of Instance in OpenStack

Fig. 3 shows the process of creating instances in Nova-Network OpenStack environment. OpenStack consists of the controller and compute nodes which can be expanded according to the size of cloud environment. Nova Scheduler in the controller node chooses a host to create instances virtualized and the scheduling algorithm which considers the entire performance is applied to the host. The host is selected among the compute nodes. When the host is determined, Nova Compute service creates instances with the memory and volume size information that users set up. Once the instance is created, Nova-Network builds up both internal and external network. IP and MAC address information of the created instance is automatically collected in the controller node. The information is then stored in the database and is used for the construction of reliable ARP table.

B. Reliable ARP Table

OpenStack consists of 8 detailed projects. Among them Keystone project is responsible for the authentication of all the services in OpenStack [7]. External users must go through the Keystone authentication to access the service in OpenStack [8]. Keystone authentication is used to construct the table, external users can only access to the table through the authentication process. So, reliability of the table is highly increased.

C. Comparison Handler

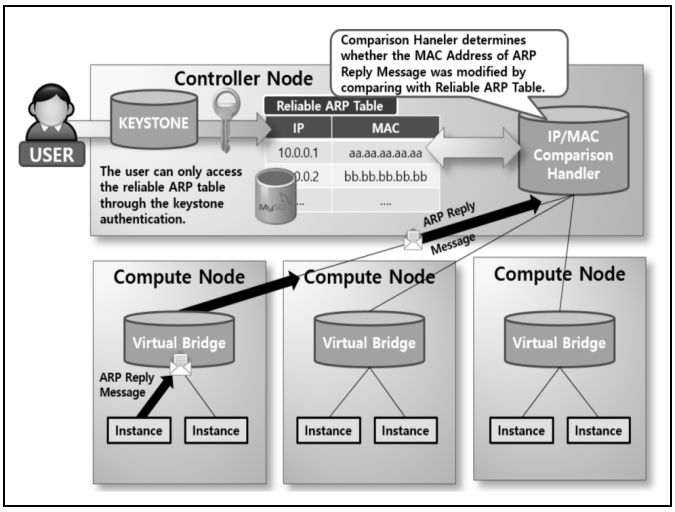

Fig 4. ARP Reply Message Handling Process using Comparison Handler

Fig. 4 shows the process of ARP Reply Message handling. Comparison Handler compares IP and MAC address information in the Reply Message to those in the reliable ARP table registered in Keystone authentication service. If the addresses are not matched, it considers the message is modified. So, Spoofing can be detected and the system is protected from the attack.

V. EVALUATION

In this paper, we assume that the attacker is a host in the same network where OpenStack nodes are located and the victim is trying to communicate with the internal instance of cloud is an external host. For the evaluation the attacking tool we choose is Cain & Abel which is well known ARP Spoofing hacking tool [9]. Since the tool can be operated only in Windows, the attacker host PC is set up with Windows 7 and all the other nodes and created instances are set up with Ubuntu 14.04 LTS.

A. Spoofing Attack Scenario

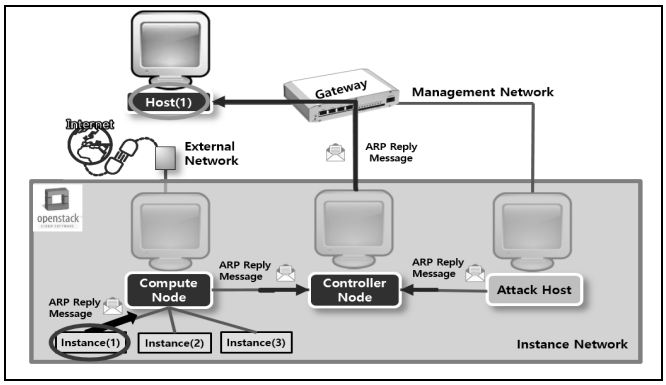

Fig 5. Spoofing Attack Scenario

Fig. 5 shows the scenario for evaluation of the proposed technique. Host(1) checks if MAC address of Instance(1) which is cached in its ARP table to communicate with Instance(1) in OpenStack. When there is no corresponding address, Host(1) broadcasts ARP Request Message to the subnet it is belonging to. After Instance(1) receives the request, it unicasts ARP Reply Message including its own MAC address to Host(1). At the same tim

e the attacker host sends the replying to Host(1). In the same way that Instance(1) does. Before these two messages arrive at Host(1), these are stopped by the controller and undergo the Comparison Handler in the controller node. If the proposed technique is implemented properly, only ARP Reply Message from Instance(1) is delivered to Host(1) whereas the other replies are dropped.

B. Spoofing Attack Results

```
Interface: 163.180.116.63 --- 0x3
 Internet Address      Physical Address    Type
 163.180.116.1         c4-7d-4f-73-a6-7f   dynamic
 163.180.116.27        d0-50-99-12-84-fc   dynamic
 163.180.116.28        00-e0-4c-36-e6-7d   dynamic
 163.180.116.26        50-46-5d-73-3f-bf   dynamic
```

Fig 6. ARP Table of Host(1) before ARP Spoofing

```
Interface: 163.180.116.63 --- 0x3
 Internet Address      Physical Address    Type
 163.180.116.1         c4-7d-4f-73-a6-7f   dynamic
 163.180.116.27        00-e0-4c-36-e6-7d   dynamic
 163.180.116.28        00-e0-4c-36-e6-7d   dynamic
 163.180.116.26        50-46-5d-73-3f-bf   dynamic
```

Fig 7. ARP Table of Host(1) after ARP Spoofing

```
Interface: 163.180.116.63 --- 0x3
 Internet Address      Physical Address    Type
 163.180.116.1         c4-7d-4f-73-a6-7f   dynamic
 163.180.116.27        d0-50-99-12-84-fc   dynamic
 163.180.116.28        00-e0-4c-36-e6-7d   dynamic
 163.180.116.26        50-46-5d-73-3f-bf   dynamic
```

Fig 8. ARP Table of Host(1) after Applying The Proposed Technique

Fig. 6 shows the ARP table of Host(1) before ARP Spoofing. In the figure 163.180.116.63, 163.180.116.1, 163.180.116.27 , 163.180.116.28 and 163.180.116.26 represents the IP address of Host(1), Gateway, Instance(1), Attack Host and Compute Node respectively. Also MAC address corresponding to each IP address can be described in the figure. Fig. 7 shows the ARP table of Host(1) after the attack has been made without any defense technique. It is possible to see that MAC address of Instance(1) is changed to MAC address of Attack Host because of Spoofing. Fig. 8 shows the table after the attack. In this case, the proposed defense technique is applied to the system. So, we can see the table remains the same as it is before the attack.

VI. Conclusion and Future Work

The technique we propose in this paper uses Keystone authentication service provided by OpenStack itself so there is no need for any additional equipments or construction. Moreover, all ARP Reply Messages are received by the controller and processed through Comparison Handler. This requires simple addition of compute nodes so it is not a burden on the cloud computing system. However if the attack is targeting either gateway or compute node that provides hypervisor service, the proposed technique is difficult to be realized. Besides, one of the most important thing in cloud computing system is resource management but it is the fact that there are partial loss of resources to maintain Comparison Handler and the table, which might be considered as a limitation of the proposed technique. Since 1982 when drawbacks of ARP were discovered there has been many types of attacks using such drawbacks. This means that the solution to the problem has been impractical or non-scalable. Even though the technique proposed in this paper has some limitations, it shows the possibilities to overcome the limitations that other previous works have. So, it is improved and possible to use in real cloud environment in near future.

VII. Acknowledgement

This work was supported by the ICT R&D program of MSIP/IITP.[R0126-15-1009, Development of Smart Mediator for Mashup Service and Information Sharing among ICBMS Platform]. *Dr. CS Hong is the corresponding author.

References

[1] Gartner, “The Top 10 Strategic Technology Trends for 2015”, 2014. 08.

[2] CSA, “The Notorious Nine – Cloud Computing Top Threats in 2013”, 2013. 02.

[3] Balachandra Reddy Kandukuri, Remakrishna Paturi V, Atanu Rakshit, “Cloud Security Issues”, 2009 Services Computing Conference, pp.517-520, 2009. 10.

[4] G. Gouda and H. Chin-Tser, "A Secure Address Resolution Protocol," Computer Networks, vol.1, no.41, pp.57-71, 2003.

[5] Michael Jarschel, Simon Oechsner, Daniel Schlosser, Rastin Pries, Sebastian Goll, Phuoc Tran Gia, “Modeling and Performance evaluation of an OpenFlow architecture”, ITC ’11 Proceedings of the 23rd International Teletraffic Congress, pp.1-7, 2011.

[6] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, Jonathan Turner, “OpenFlow: Enabling Innovation in Campus Networks”, ACM SIGCOMM Computer Communication Review, vol.38, pp.69-74, 2008

[7] http://docs.openstack.org/icehouse/install/guide/install/apt/content/ch_overview.html

[8] Rasib Hassan Khan, Jukka Ylitalo, Abu Shohel Ahmed, “OpenID authentication as a service in OpenStack”, 2011 7th Information Assurance and Security (IAS), pp.372-377, 2011. 12.

[9] Cain & Abel, http://www.oxid.it/projects.html