OpenStack 클라우드 컴퓨팅 환경에서 Keystone 인증서비스를 이용한 ARP Spoofing 방어기법

강효성○ 홍충선
경희대학교 컴퓨터공학과
{ kanghs○, cshong } @khu.ac.kr

Defense Technique against ARP Spoofing Attacks using Keystone Authentication Service in OpenStack Cloud Computing Environment

Hyo Sung Kang○ Choong Seon Hong
Department of Computer Engineering, Kyung Hee University

요 약

최근 많은 전 세계 기업들은 비용절감, 업무프로세스의 혁신을 목적으로 클라우드 서비스를 도입하고 있지만 클라우드 내 가상머신을 대상으로하는 외부 Spoofing 또는 Poison 공격으로 인해 발생 될 수 있는 클라우드 시스템의 성능 저하는 클라우드 컴퓨팅 시스템 확산의 많은 걸림돌이 되고 있다. 이런 보안사고를 예방하기위해 많은 연구가 진행되었지만 실현가능성이 낮은 새로운 프로토콜의 제안, 대규모 네트워크에 적용하기 어려운 확장성의 한계를 가지고 있었다. 이에 본 논문에서는 최근 많은 클라우드 서비스 개발 회사가 차용하고 있는 오픈소스 클라우드 플랫폼, OpenStack을 이용하여 클라우드 환경을 구축하고 OpenStack의 인증서비스를 제공하는 Keystone을 이용한 인증 테이블을 통하여 Spoofing 혹은 Poison이라 불리는 네트워크 공격에 대한 방어기법을 제안하고자 한다.

1. 서 론

전 세계 많은 기업들은 IT 비용에 대한 원가절감, 업무프로세스의 혁신을 목적으로 클라우드 컴퓨팅 서비스를 도입하고 있다. 미국 내(內)의 IT 컨설팅 제공업체인 가트너(Gartner)는 "2015년 10대 전략기술(The Top 10 Strategic Technology Trends for 2015)" [1]에서 7위로 클라우드 컴퓨팅 기술을 언급하며 현재 클라우드 컴퓨팅 기술은 IT 핵심 전략기술로 급부상되고 있다. 그러나 아직도 다수의 기업들은 실제 클라우드 컴퓨팅 서비스 도입을 망설이고 있는데 그 중 가장 큰 장애요소는 무엇보다 보안에 대한 우려이다.

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 Vulnerabilities

표 1. 2015년 클라우드 컴퓨팅에서의 9가지 위협 요소

표 1은 국제클라우드보안협회(CSA, Cloud Security Alliance)에서 소개한 클라우드 컴퓨팅에서 가장 위협이 되는 9가지 요소들이다[2]. 이 중 데이터의 유출과 손실(Data Breaches & Loss) 그리고 계정 또는 서비스 트래픽 가로채기(Account or Service Traffic Hijacking)와 같은 위협요소들은 개인정보의 유출 뿐만 아니라 하나의 내부 네트워크에서 여러 사용자에게 호스팅 서비스를 제공하는 클라우드 환경 특성상 전체 클라우드 시스템의 성능 저하의 원인이 된다. 그리고 이런 위협들은 흔히 Spoofing 또는 Poison이라고 부르는 외부 네트워크 공격으로 쉽게 구현 될 수 있다. 이런 보안적 위협요소를 예방하기 위해 많은 연구가 진행되었지만 기존의 연구에서는 실현가능성이 낮은 새로운 프로토콜의 제안[3], 대규모 클라우드 네트워크 환경에 적용하기 어려운 확장성[5]의 한계를 가진다.

이에 본 논문에서는 외부 네트워크 공격 기법 중 가장 많이 사용되고 있는 ARP Spoofing 공격에 대해 오픈소스 클라우드 플랫폼, OpenStack을 이용하여 클라우드 컴퓨팅 환경을 구축하고 OpenStack의 인증 서비스인 Keystone을 이용하여 인증 테이블을 생성해 이를 활용한 외부 Spoofing 공격 방어기법을 제안하고자 한다.

2. ARP and ARP Spoofing

ARP(Adress Resolution Protocol)는 네트워크 계층의 주소를 데이터 링크 계층 주소로 변환하는 표준 프로토콜로 특정 데이터를 목적지까지 전송하기 전에 목적지 IP 주소에 해당하는 MAC 주소가 ARP 테이블에 없을 경우 이를 알아내기 위해 사용한다. 여기서 ARP는 별다른 인증 메커니즘 없이 네트워크에 있는 단말장치의 MAC 주소 정보를 확인할 수 있다는 보안적 문제를 안고 있는데 이런 취약점을 이용한 대표적인 네트워크 공격기법이 ARP Spoofing이다. ARP Spoofing은 공격 대상자에게 조작된 ARP Reply Message를 전달해 특정 호스트의 MAC 주소를 잘못 인식하게 하여 정상적인 통신을 방해하는 공격방식이다.

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R0126-15-1009, ICBMS 플랫폼 간 정보 모델 연동 및 서비스 매쉬업을 위한 스마트 중재 기술 개발]. *Dr. CS Hong is the corresponding author

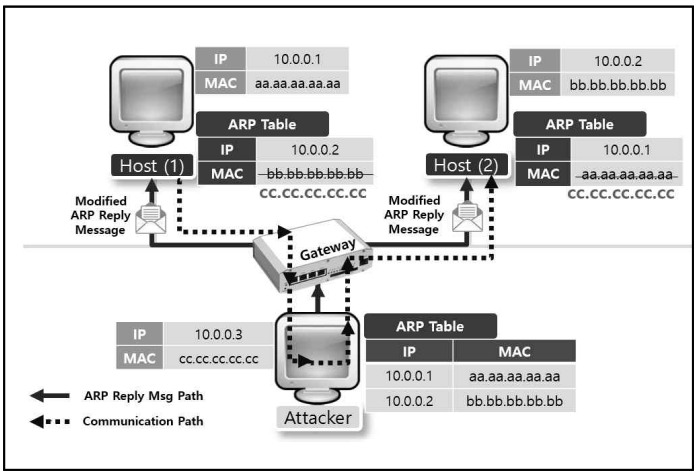

그림 1. ARP Spoofing Attack

그림 1은 이런 ARP Spoofing 공격 과정을 보여준다. Host(1)은 Host(2)와 통신을 하기 위해 Host(2)의 MAC 주소를 알아야하며 마찬가지로 Host(2)도 Host(1)과 통신을 하기 위해 Host(1)의 MAC 주소를 알아야한다. 이를 위해 Host(1)과 Host(2)는 자신이 속해있는 서브넷(Subnet)에 ARP Request Message를 브로드캐스트한다. 이때 같은 네트워크상에 존재하는 Attacker는 Host(1)과 Host(2)의 Request Message를 인지하고 Host(1)과 Host(2)에 자신의 MAC 주소를 가지는 조작된(Modified) ARP Reply Message를 보내어 Host(1)과 Host(2)의 ARP 테이블을 조작한다. 결과적으로 Host(1)은 Attacker를 Host(2)로 인식하고, Host(2)는 Attacker를 Host(1)로 인식하여 Host(1)과 Host(2)는 서로가 통신을 하고 있다고 믿지만 실제로는 Attack와 통신을 하며 정보를 Attacker에게 노출하는 결과를 낳는다.

3. 관련 연구

3.1 Secure Address Resolution Protocol

ARP Spoofing의 가장 큰 원인은 ARP Request에 대한 별다른 인증없는 ARP Reply Message 수신과정이다. 이를 개선하기 위해 제한된 것이 SARP(Secure Address Resolution Protocol)로 ARP Reply 메시지 수신과정에 인증절차를 추가하여 새로운 프로토콜로 정의한 것이다[3]. 해당 알고리즘은 ARP Spoofing 공격의 근본원인을 해결할 수는 있지만 이미 널리 사용 중인 프로토콜을 새로운 프로토콜로 대체한다는 것은 현실적으로 불가능하다.

3.2 OpenFlow를 이용한 Spoofing 방지 메커니즘

오픈플로는 네트워크 스위치 또는 라우터의 Forwarding Table을 원격에서 제어할 수 있는 접속권한을 제공하는 통신 프로토콜이다[4]. 여기서는 OpenFlow의 Controller에 네트워크에 연결된 모든 단말장치의 IP, MAC 주소 정보를 등록하고 ARP Request Message에 대해 일차적으로 Controller가 우선 수신하도록 설정한 뒤 미리 등록되어 있는 IP, MAC 주소와 비교하여 일치하지 않는 ARP Reply Message를 제거(drop) 하는 방식이다[5]. 현재 OpenFlow는 자체 제공되는 API를 통해 네트워크망을 동적으로 관리할 수 있다는 장점으로 많은 클라우드 네트워크 구축에 활용되고 있다. 그렇기 때문에 해당 메커니즘의 구현도 쉽고 클라우드 환경에 적용도 쉽다는 장점이 있지만 OpenFlow가 적용되지 않은 클라우드 컴퓨팅 환경에는 적용하기 어려워 확장성이 낮은 단점이 존재한다[5].

4. 제안하는 방어기법

OpenStack은 오픈소스 클라우드 플랫폼으로 Network 서비스의 단독 구성 여부에 따라 두가지 설치구성을 제안한다[6]. 본 논문에서는 TCP/IP 기반의 클라우드 환경에서의 방어기법을 제안하는 것이므로 단독 구성이 아닌 네트워크 서비스가 인스턴스 서비스에 포함된 Nova-Network를 이용한 OpenStack 구성을 이용한다.

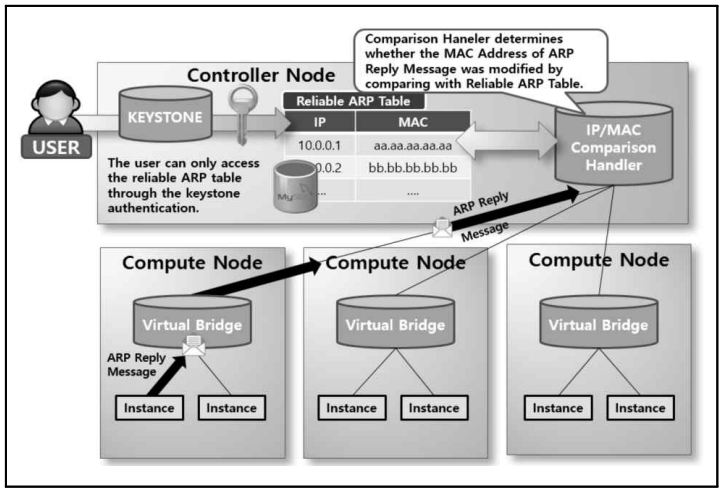

그림 2. OpenStack에서 인증테이블을 이용한 ARP Reply Message 처리 과정

그림 2는 본 논문에서 제안하는 OpenStack 클라우드 환경에서 인증테이블을 이용한 ARP Reply Message의 처리 과정이다. 제안된 기법은 OpenStack 컴퓨팅 클라우드 환경에서 생성된 인스턴스들의 IP, MAC 주소 정보를 수집하여 OpenStack의 Keystone을 이용한 인증 테이블을 생성, 관리하는 부분과 ARP Reply Message의 처리를 담당하는 비교처리기(Comparision Handler)부분으로 구성된다. 우선 Controller Node는 Compute Node에 생성된 각 인스턴스의 IP, MAC 주소정보를 수집한다. 수집된 OpenStack 프로젝트 자체에서 제공하는 Keystone 인증 서비스를 통해 인증 테이블을 구성하고 Controller Node에 저장된다.

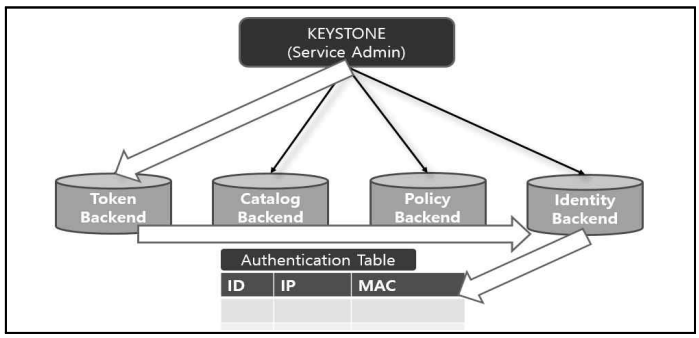

그림 3. Keystone을 이용한 인증테이블 생성

그림 3은 Keystone을 이용한 인증테이블을 생성하는 과정이다. Keystone은 수집된 IP, MAC 정보에 각각 Token Backend라는 서비스

를 통해 인증 ID를 생성하고 생성된 ID 정보의 인증절차를 담당하는 Identity Backend 서비스를 통해 인증테이블을 생성한다. 그리고 여기서 브로드캐스트 되는 ARP Request Message에 대한 Reply Message를 우선 Controller Node를 거치게 하여 인증테이블과 비교 후 IP, MAC 주소가 일치하지 않는 경우에는 조작된 ARP Reply Message로 간주해 차단함으로써 원천적인 Spoofing 공격을 방지한다.

4. 성능평가

4.1 ARP Spoofing 공격 시나리오

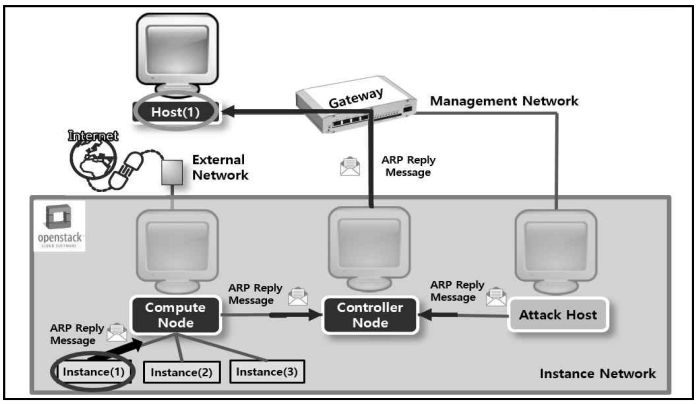

그림 4. 성능 평가를 위한 Spoofing Attack 시나리오

그림 4는 본 논문에서 제안한 방어기법의 성능평가를 위한 Spoofing 공격 시나리오이다. Host(1)은 OpenStack 내부 Instance(1)과 통신을 하기 위해 목적지 Instance(1)의 MAC주소가 자신의 ARP 테이블에 캐싱되어 있는지 확인한다. 캐싱된 주소가 없으면 Host(1)은 자신이 속해있는 서브넷(Subnet)에 ARP Request Message를 브로드캐스트한다. ARP Request Message를 수신한 Instance(1)은 자신의 MAC 주소가 포함된 ARP Reply Message를 Host(1)로 유니캐스트하고 공격자 호스트는 자신의 MAC 주소로 조작된 ARP Reply Message를 Host(1)로 유니캐스트 한다. 2개의 Reply Message는 Host(1)에 도달하기 전에 우선 Controller Node의 비교차단기를 거치게 되는데, 제안된 기법이 올바로 동작한다면 인증테이블과의 비교를 통해 Instance(1)의 ARP Reply Message는 Host(1)에 전달될 것이고 공격자 Host의 ARP Reply Message는 차단 될 것이다.

4.2 성능 평가

```
인터페이스: 163.180.116.63 --- 0x3
  인터넷 주소           물리적 주소           유형
  163.180.116.1         c4-7d-4f-73-a6-7f     동적
  163.180.116.27        d0-50-99-12-84-fc     동적
  163.180.116.28        00-e0-4c-36-e6-7d     동적
  163.180.116.26        50-46-5d-73-3f-bf     동적
```

그림 5, ARP Spoofing 공격전의 Host(1)의 ARP Table

```
인터페이스: 163.180.116.63 --- 0x3
  인터넷 주소           물리적 주소           유형
  163.180.116.1         c4-7d-4f-73-a6-7f     동적
  163.180.116.27        00-e0-4c-36-e6-7d     동적
  163.180.116.28        00-e0-4c-36-e6-7d     동적
  163.180.116.26        50-46-5d-73-3f-bf     동적
```

그림 6, ARP Spoofing 공격후의 Host(1)의 ARP Table

```
인터페이스: 163.180.116.63 --- 0x3
  인터넷 주소           물리적 주소           유형
  163.180.116.1         c4-7d-4f-73-a6-7f     동적
  163.180.116.27        d0-50-99-12-84-fc     동적
  163.180.116.28        00-e0-4c-36-e6-7d     동적
  163.180.116.26        50-46-5d-73-3f-bf     동적
```

그림 7, 방어기법 적용 후의 Host(1)의 ARP Table

그림 5는 ARP Spoofing 공격전의 Host(1)의 ARP Table을 보여준다. 여기서 163.180.116.1은 게이트웨이이고 163.180.116.27은 Instance(1)의 IP, 163.180.116.26은 Compute Node의 IP, 163.180.116.63은 Host(1)의 IP 그리고 163.180.116.28은 Attack Host의 IP 주소이며 각 IP 주소에 대응되는 MAC 주소를 확인할 수 있다. 그림 6은 제안된 방어기법 없이 일반적인 Spoofing 공격을 했을 때 ARP 테이블이다. ARP Spoofing 공격으로 인해 Instance(1)의 MAC 주소가 Attack Host의 MAC주소로 변경된 것을 확인 할 수 있다. 그리고 그림 7은 제안된 기법을 적용한 뒤에 ARP Spoofing 공격 후 Host(1)의 ARP Table이다. ARP Spoofing 공격 전의 ARP Table을 유지하는 것을 확인할 수 있다.

5. 결론

본 논문에서 제안된 기법은 OpenStack 자체 서비스에서 제공하는 Keystone 인증서비스를 이용하기 때문에 외부에 추가적인 장비나 구성이 필요없다. 또한 Controller Node에서 전체 ARP Reply Message를 수신받아 Comparison Handler를 통해 처리하기 때문에 단순히 Compute Node를 더해나가는 Cloud 컴퓨팅 시스템의 확장에도 큰 무리를 주지 않는다. 하지만 공격자 호스트가 처음부터 Gateway 혹은 하이퍼바이저 서비스를 제공하는 Compute Node를 공격대상으로 삼았다면 제안된 방어기법은 적용하기가 힘들다. 또한 클라우드 컴퓨팅 시스템에서 가장 중요한 요소가 바로 자원의 관리인데 인증테이블과 Comparison Handler를 유지하는데 일부 자원이 낭비되는 것도 제안된 기법의 한계이다. 제안한 방어기법은 과거에 보여주었던 여러 연구들이 가진 단점들의 극복가능성을 보여주었다. 앞으로 조금만 더 개선된다면 실제 네트워크 환경에 적용도 용이할 것이라고 전망한다.

참고문헌

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

[4] 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.

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