Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.3, Jun. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.3.579

클라우드 환경에서의 악성트래픽 동적 분석 시스템 설계*

이 은 지,[1†] 곽 진[2‡]
[1]아주대학교 컴퓨터공학과 정보보호응용및보증연구실
[2]아주대학교 사이버보안학과

Design of Malicious Traffic Dynamic Analysis System in Cloud Environment*

Eun-Ji Lee,[1†] Jin Kwak[2‡]
[1]ISAA Lab., Department of Computer Engineering, Ajou University
[2]Department of Cyber Security, Ajou University

요 약

클라우드 환경은 하이퍼바이저 기반으로 다수의 가상머신들이 상호 연결된 형태로 악성코드의 전파가 용이하기 때문에 다른 환경에 비해 악성코드에 감염될 경우 그 피해규모가 상대적으로 크다. 본 논문에서는 이러한 문제점을 해결하기 위해 안전한 클라우드 환경을 위한 악성트래픽 동적 분석 시스템을 제안한다. 제안하는 시스템은 클라우드 환경에서 발생하는 악성트래픽을 판별하여 악성행위를 격리된 가상네트워크 환경에서 지속적으로 모니터링 및 분석한다. 또한, 분석된 결과를 추후 발생하는 악성트래픽의 판별과 분석에 반영한다. 본 논문에서 제안하는 시스템은 클라우드 환경에서 발생하는 신·변종 악성트래픽 탐지 및 대응을 목적으로 클라우드 환경에서의 악성트래픽 분석환경을 구축함으로써 안전하고 효율적인 악성트래픽 동적 분석을 제공한다.

ABSTRACT

The cloud environment is hypervisor-based, and many virtual machines are interconnected, which makes propagation of malicious code easier than other environments. Accordingly, this paper proposes a malicious traffic dynamic analysis system for secure cloud environment. The proposed system continuously monitors and analyzes malicious activity in an isolated virtual network environment by distinguishing malicious traffic that occurs in a cloud environment. In addition, the analyzed results are reflected in the distinguishment and analysis of malicious traffic that occurs in the future. The goal of this research is secure and efficient malicious traffic dynamic analysis by constructing the malicious traffic analysis environment in the cloud environment for detecting and responding to the new and variant malicious traffic generated in the cloud environment.

Keywords: Cloud, Malware Analysis, Security

I. 서 론

2015년 '클라우드컴퓨팅 발전 및 이용자 보호에 관한 법률' 시행으로 국내 공공분야의 클라우드 도입 기반이 마련되고 있다. 또한, 미국, 영국, 일본 등의 주요국에서는 클라우드를 도입함에 있어 보안성 강화를 강조하는 등 클라우드 보안의 중요성이 대두되고 있다[1].

특히, 클라우드 환경은 가상화 기술을 기반으로 하이퍼바이저를 통해 다수의 가상머신들이 상호 연결되어있어 악성코드 전파가 용이하다[2]. 따라서, 클라우드 환경에서 악성트래픽이 발생할 경우 다수의

Received(03. 14. 2017), Modified(04. 28. 2017), Accepted(04. 29. 2017)
* 본 논문은 2016년도 동계학술대회에 발표한 우수논문을 개선 및 확장한 것임
† 주저자, heo160@ajou.ac.kr
‡ 교신저자, security@ajou.ac.kr(Corresponding author)

가상머신들이 동시에 피해를 볼 수 있기 때문에 다른 환경에 비해 피해 규모가 크다.

이러한 이유로, 시만텍(Symantec), 안랩(AhnLab) 등의 기업들에서는 클라우드 보안을 위한 악성트래픽 관제 및 분석 연구가 활발히 진행 중에 있으며[3,4], 이와 함께 일반적인 컴퓨팅 환경에서 새로운 유형의 악성트래픽을 탐지하고 악성행위를 분석하기 위한 악성트래픽 동적 분석 시스템 설계 방안에 대한 연구 사례 또한 존재한다[5].

이에 따라, 본 논문은 안전한 클라우드 환경을 위한 악성트래픽 동적 분석 시스템을 제안한다. 제안하는 시스템은 클러스터링을 이용한 악성트래픽 분석 및 판별, 격리환경에서 재전송 공격을 통한 지속적인 악성행위 모니터링 기술을 포함한다. 이는 클라우드 환경에서 발생하는 악성트래픽의 동적 분석을 목표로 한다.

본 논문의 구성은 다음과 같다. 2장에서 기존 악성트래픽 동적 분석 시스템, OvS 컨트랙 기능 및 본 논문에서 선정한 악성코드 샘플인 Zeus 봇넷의 통신 과정을 분석한다. 3장에서는 클라우드 환경에서의 악성트래픽 동적 분석 시스템을 제안하고, 4장에서 실험 결과 및 분석을 진행하며, 5장에서 결론을 맺는다.

II. 관련연구

2.1 악성트래픽 동적 분석 시스템

Mariano Graziano 등[5]은 일반적인 컴퓨팅 환경에서 새로운 유형의 진화된 악성코드 탐지를 위한 악성코드 동적 분석 시스템을 제안하였다. 이는 인터넷 호스트와 악성코드 샘플의 상호작용에 의존하는 샌드박스 및 필터링 정책을 이용하는 기존의 악성코드 탐지 기술의 한계를 개선한 시스템이며, 시스템 구성은 Fig.1.과 같다.

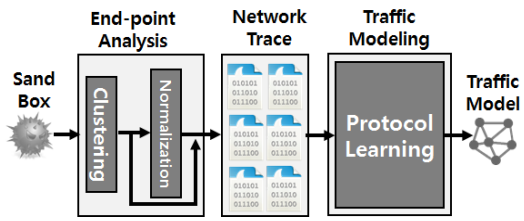

Fig. 1. Dynamic Analysis system of Mariano Graziano et al.[5].

제안된 시스템은 트래픽 수집(Traffic Collection)단계, 엔드포인트 분석(Endpoint Analysis)단계, 트래픽 모델링(Traffic Modeling)단계, 트래픽 봉쇄(Traffic Containment)단계의 4단계로 구성되며, 각 단계에 대한 설명은 아래와 같다.

Step1. 트래픽 수집

악성코드 샘플 실행과 관련하여 네트워크 트레이스(trace)를 수집한다. 이때, 샌드박스나 기존의 분석결과를 바탕으로 악성코드를 수집하거나 트레이스를 추출할 수 있다.

Step2. 엔드포인트 분석

수집된 트레이스를 정규화하는 단계이다. 이는 클러스터링을 통해 IP fluxing 등을 이용하는 악성코드 트레이스를 군집화 및 정규화 시킨다. 이 단계의 결과는 트래픽 수집단계에 지속적으로 영향을 준다.

Step3. 트래픽 모델링

이 단계는 프로토콜 러닝(learning) 알고리즘을 통해 수집된 트래픽의 IP 주소, 포트 등을 상호 연결하여 악성코드와 엔드포인트 간 상호작용 모델링을 구성한다.

Step4. 트래픽 봉쇄

안전한 악성코드 분석환경을 구축하기 위해 트래픽이 외부와 통신하지 못하도록 완전봉쇄를 이루는 단계이다. 이때, 프록시 기술을 이용하여 외부와 통신하는 트래픽을 리다이렉션(redirection) 시키는 방식을 이용한다.

구축된 분석환경에서 전 단계들을 지속적으로 수행하여 악성코드 분석 정보를 개선시킨다.

2.2 Zeus 봇넷 통신 과정

본 논문은 초기 악성코드 분석 샘플로 Zeus 봇넷을 이용하며, Zeus 봇넷 분석을 위한 Zeus 봇넷과 봇의 네트워크 통신 과정은 아래와 같다.

Zeus 봇넷과 봇은 암호화된 HTTP 프로토콜을 사용한다. 봇을 통해 사용자 PC를 감염시키면 봇은 봇넷 서버에 환경설정 파일인 config.bin*을 GET으로 요청한다. 봇넷은 암호화된 환경설정 파일과 함께 해당 요청에 대한 응답패킷 200 OK를 봇에게

전송한다[6].

이후, 봇은 봇넷에 사용자 PC의 정보를 유출하는 gate.php 파일을 GET으로 전송하고 봇넷은 이에 대한 응답패킷 200 OK를 전송한다. 이와 같이 봇넷과 봇 간 초기 통신이 이루어지면 봇넷은 스크립트를 통해 공격 명령을 수행하며, 공격 명령을 받은 봇은 해당 공격 명령이 요구하는 사용자 정보를 포함하는 gate.php를 POST로 전송한다. 이때 gate로 명시된 파일 이름은 스크립트를 통해 임의로 수정이 가능하기 때문에 *.php 파일로 명시할 수 있다. 이에 따른 Zeus 봇넷의 네트워크 통신 과정은 Fig.2.와 같다[6].

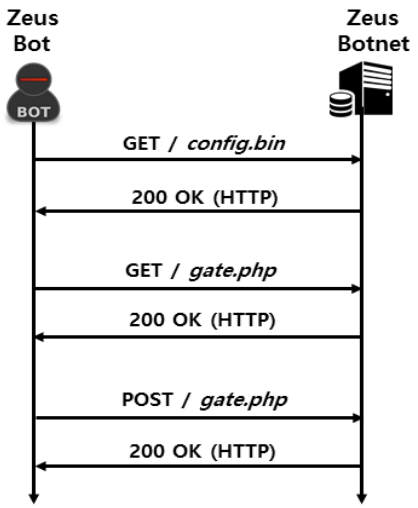

Fig. 2. Communication pattern of Zeus

2.3 OvS 컨트랙

클라우드 환경을 제공하는 오픈스택(openstack)은 OvS(OpenvSwitch)를 통해 가상머신 간 패킷 스위칭, 제어 등의 핸들링 기능을 지원한다. OvS는 하이퍼바이저 기반의 오픈소스 가상 스위치로 유저 공간(user space)과 커널(kernel)로 구분된다. 유저 공간에서는 패킷 스위칭 등의 핸들링 명령을 제공하며, 커널에서는 해당 명령을 받아 플로우 테이블(Flow Table)을 이용하여 실질적인 패킷 핸들링 동작을 수행한다[7].

컨트랙(conntrack)은 넷필터(netfilter) 프레임워크에서 제공하는 기능으로 네트워크의 연결 추적 정보를 알아낼 수 있다. 2015년 이후 배포된 OvS 2.5 버전부터 OvS에 컨트랙 기능이 추가되었으며[8], 이는 기존의 stateless 패킷처리방식의 한계를 개선한 stateful 패킷처리방식을 제공한다.

기존의 stateless 방식은 전송되는 해당 패킷에만 영향을 받으며 IP주소, 포트 등 L3 및 L4의 네트워크 정보를 이용해 패킷을 검사하여 처리한다. 하지만, 최근 방화벽과 같은 보안 기능 관점에서는 기존의 방식보다 더욱 발전된 패킷처리 및 필터링 방식을 요구한다. 이를 위해, statefull 방식은 패킷의 네트워크 커넥션(connection) 정보를 검사하여 패킷을 처리한다. 커넥션 정보는 초기 패킷이 전송될 때 패킷의 헤더, 페이로드의 L7 정보 등을 포함한다. 이는 네트워크가 연결될 때의 연결 상태를 추출한 정보로 기존의 L3, L4 헤더 정보보다 자세한 정보를 통해 패킷을 처리할 수 있다. 이러한 방식은 넷필터의 컨트랙 기능을 이용해 가능하다[9].

OvS 컨트랙은 네트워크 연결이 시도될 때 커넥션 정보와 같은 플로우(flow) 정보를 이용해 패킷을 처리한다. 플로우 정보는 컨트래커(conntracker)가 기억하도록 하여 기억된 정보를 통해 추후 전송되는 패킷을 커널에서 처리가 가능하다. 이에 따른 OvS 컨트랙 구성은 Fig.3.과 같고 동작 단계는 아래와 같다[9].

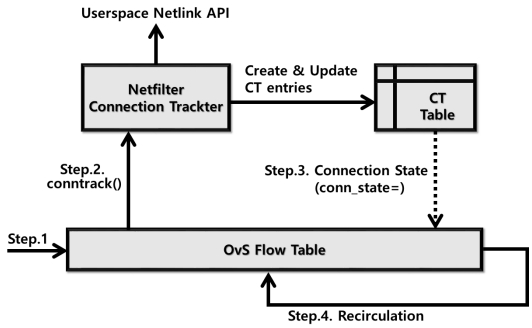

Fig. 3. Composition of OvS conntrack

Step1. 패킷이 OvS를 통해 플로우 테이블에 전송된다.

Step2. 넷필터 컨트래커에 패킷이 전송된다.

Step3. 커넥션 테이블(CT Table)은 플로우와 TCP 윈도우 크기를 적용하고, OvS 플로우의 커넥션 상태(conn_state)를 가능하게 하며, 커넥션 비

트 셋(bits set)과 함께 OvS 플로우 테이블에 다시 전송된다.

Step4. Recirculation 옵션을 사용하면, OvS 플로우 테이블에 저장된 기존 커넥션 정보를 이용하여 컨트랙이 항상 커널에 생성된 첫 번째 action을 수행한다.

격리된 분석환경에서 악성트래픽을 모니터링하기 위해 악성행위를 지속적으로 발현시킬 필요가 있다. 이에 따라, 본 논문은 OvS 컨트랙을 이용해 악성트래픽을 핸들링하여 재전송 공격을 일으킴으로써 지속적으로 악성행위를 발현시키는 방안을 제안한다.

III. 제안기법

3.1 제안하는 동적 분석 시스템

본 논문은 Mariano Graziano 등[5]이 악성트래픽 동적 분석을 위해 제시한 필요 사항을 기반으로 클라우드 환경에 적합한 악성트래픽 동적 분석 시스템을 제안한다. 본 논문이 제안하는 클라우드 환경에서의 동적 분석 시스템은 악성트래픽 분석단계, 격리환경 구축단계, 악성행위 발현단계의 3단계로 구성된다. 이에 따른 동적 분석 시스템은 Fig.4.와 같으며, 시스템의 각 단계별 설명은 아래와 같다.

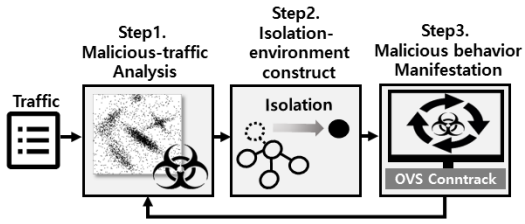

Fig. 4. Proposed dynamic analysis system

Step1. 악성트래픽 분석

악성트래픽 분석단계는 호스트 가상머신으로 전송되는 트래픽을 클러스터링하여 악성트래픽을 판별하고 분석하는 단계이다. 클러스터링 기술을 통해 악성트래픽의 악성행위를 유형별로 군집화함으로써 분석된 정보를 기반으로 악성트래픽을 판별한다.

본 논문은 클러스터링 알고리즘으로 k-means 알고리즘을 이용하며 초기 악성트래픽 샘플은 Zeus 봇넷으로 선정하여 연구를 진행하였다.

Step2. 격리환경 구축

악성트래픽 분석단계에서 판별된 악성트래픽의 동적 분석을 위해 악성행위 모니터링하기 위한 격리환경을 구축하는 단계이다. 이때, 외부 C&C 서버와의 통신, 자료유출 등 악성트래픽 특성에 의해 외부와 통신이 불가한 격리된 환경에서 악성행위를 발생 및 모니터링할 필요가 있다.

격리환경 구축은 마이그레이션, 볼륨 이동 등과 같은 방식으로 가능하다. 마이그레이션은 시스템 복구, 유지 등의 목적으로 해당 가상머신을 다른 가상머신으로 복사하는 기술이다[10]. 더불어, 볼륨 이동은 가상머신에 연결된 볼륨을 다른 가상머신에 이동시킴으로써 해당 가상머신을 복사하는 방식이다.

Step3. 악성행위 발현

이 단계는 격리된 환경에서 악성행위 모니터링을 위해 악성행위를 발현시키는 단계이다. 격리된 환경에서는 악성트래픽이 C&C 서버와 통신이 불가하여 C&C 서버로부터 공격 명령을 받을 수 없기 때문에 악성트래픽이 동작하지 않는다.

그러므로, OvS 컨트랙 기능을 통해 악성트래픽을 핸들링하여 재전송 공격을 일으킴으로써 격리된 환경에서 악성행위를 일으키는 방안을 제안한다. 또한, 이 단계에서 발생하는 모니터링 정보는 악성트래픽 분석단계의 악성트래픽 클러스터링 과정에 지속적으로 영향을 미친다.

3.2 동적 분석 시스템 단계별 적용 기술

- 악성트래픽 분석

Zeus 봇넷이 제공하는 공격 명령을 통해 공격을 수행하고 이때 발생된 트래픽을 캡쳐하여 데이터 셋을 추출한다. 추출된 데이터 셋을 통해 악성행위 유형별로 트래픽을 분석하며, 분석된 각 유형의 특징을 클러스터링 알고리즘에 적용하여 악성행위 유형별 군집화한다.

악성트래픽 분석단계의 시나리오 구성은 Fig.5.와 같다.

이때, 2.2절에서 Zeus 봇넷의 통신과정을 분석한 내용을 바탕으로 악성트래픽의 추출된 데이터 셋을 분석함으로써 악성행위 유형별 특징을 도출할 수 있다. 즉, 각 공격 유형별 통신 과정에서 서로 다른 통

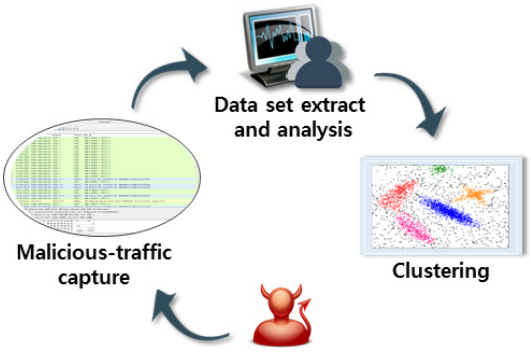

Fig. 5. Analysis step scenario of malicious traffic

신 패턴을 갖으며 실험을 통해 통신 패턴에서의 차이점을 분석하여 아래의 특징을 추출할 수 있다.

첫째로, 각 유형별로 공격 명령 시 봇이 봇넷에 전송하는 *.php에 대응하는 응답 패킷 데이터의 길이가 다른 특징을 갖는다. 봇넷이 봇에 공격 명령을 내리기 위해 *.php를 전송하고, 봇은 이에 해당하는 200 OK 응답 패킷을 전송한다. 실험 결과 공격 유형별로 해당 응답 패킷의 길이가 다른 특징을 얻을 수 있었다.

둘째로, 악성행위 유형별로 전송되는 *.php의 개수가 다른 특징을 갖는다. 봇넷이 봇에 공격 명령을 내리면 *.php 및 해당 200 OK 응답 패킷을 서로 전송하는 과정을 지속적으로 반복한다. 이때, 해당 과정을 반복하는 횟수가 공격 유형별로 다르며, 이 또한 실험 결과로써 얻어진 데이터 셋을 분석함으로써 얻을 수 있는 특징이다.

위의 도출된 특징을 이용하여 악성트래픽을 분석 및 판별할 수 있으며, 이를 위해 k-means 클러스터링 알고리즘을 통해 악성행위 유형별 군집화 하는 방안을 제안한다. k-means 클러스터링 알고리즘은 k개의 중심 값을 선정하여 새로운 데이터에 대해 각 중심 값과 거리를 비교함으로써 비슷한 거리에 근거한 군집화 기법이다. 이때, 중심 값 k는 악성행위 유형의 개수로 정의하며 위에서 도출한 특징을 알고리즘에 적용하여 군집화한다.

● 격리환경 구축

격리환경 구축단계는 클라우드 환경 구축과 격리환경 구축 과정으로 이루어진다. 클라우드 환경은 레드햇 기반의 오픈스택을 통해 구축한다. 이후, 오픈스택 내 외부와 통신이 가능한 외부 네트워크과 외부와 통신이 차단된 격리 네트워크를 구성하며 각 네트워크에 가상머신 인스턴스를 생성한다.

외부 네트워크의 인스턴스에 발생하는 트래픽이 악성트래픽 분석단계를 통해 악성트래픽으로 판별되면 해당 인스턴스는 격리 네트워크의 인스턴스로 복사된다. 이는 마이그레이션, 볼륨 이동 등을 통해 가능하다. 이에 따른 격리환경 구축단계의 시나리오 구성은 Fig.6.과 같다.

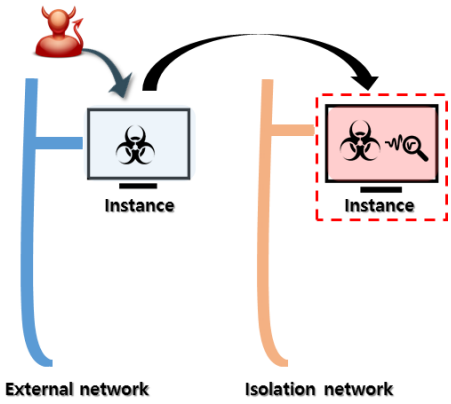

Fig. 6. Construct step scenario of isolation environment

● 악성행위 발현

외부와 통신이 차단된 격리환경에서 악성행위를 지속적으로 일으키기 위해 OvS 컨트랙 기능을 이용하여 악성트래픽 핸들링을 통한 재전송 공격 방안을 제안한다. 악성트래픽이 발생하면 OvS 커널에서 컨트랙 기능을 통해 네트워크 커넥션 정보를 추출한다. 이때, 해당 정보를 기반으로 넷필터 정책을 적용하여 OvS의 패킷처리 기능을 제어함으로써 악성트래픽의 공격 명령을 핸들링할 수 있다.

이러한 과정에서 OvS 컨트랙은 악성트래픽의 기존 네트워크 커넥션 정보를 저장한다. 이때, 해당 정보의 근원지 및 목적지 주소 또는 인터페이스 정보 등을 변경함으로써 악성트래픽 핸들링이 가능하며, 이를 통해 격리된 환경에서 재전송 공격을 일으킬 수 있다. 이에 따른 악성행위 발현단계의 시나리오 구성은 Fig.7.과 같다.

이 단계는 격리된 환경에서 악성행위를 지속적으로 발현시킴으로써 악성행위 모니터링을 가능하게 한다. 또한, 모니터링된 정보가 악성트래픽 분석단계에

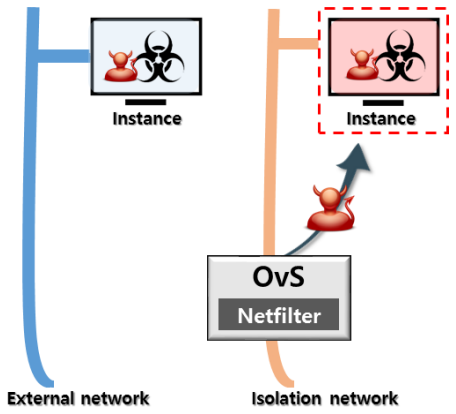

Fig. 7. Scenario of Malicious behavior Manifestation

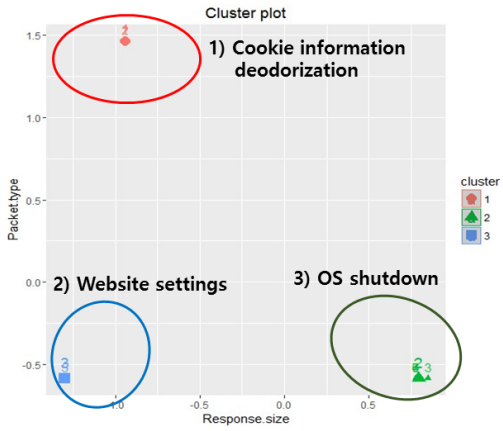

Fig. 8. Clustering result of malicious traffic

영향을 주기 때문에 악성트래픽 분석을 위한 정보가 지속적으로 업데이트될 수 있다.

IV. 실험결과 및 분석

연구를 위한 실험환경은 Table 1.과 같다.

Table 1. Experiment environment

classification	environment
OS	RedHat Enterprise Linux-7.2
kernel	Linux kernel 4.4.5
openstack	packstack Kilo
OvS	OpenvSwitch-2.5.1

4.1 악성트래픽 분석

초기 악성트래픽 분석을 위한 샘플로 Zeus 봇넷을 이용하였으며, k-means 클러스터링 알고리즘을 통해 악성행위 유형별로 군집화하였다. k-means 알고리즘은 R 코드를 통해 구현하였다. 이때, k는 악성행위 유형의 개수로 정의되며 1)웹페이지 쿠키 정보 탈취, 2)홈페이지 세팅, 3)OS 셧다운의 3가지 악성행위 유형으로 실험을 진행하였다.

k-means 알고리즘은 관측치를 기준으로 군집 중심 k에 가장 가까운 거리에 군집하는 방식이다. 따라서, 관측치는 앞서 분석한 악성트래픽 특징인 악성행위 유형별 응답패킷의 길이 및 악성행위 유형에 따른 *.php 패킷의 개수로 정의할 수 있다. 정의한 관측 변수를 임의로 각각 x축과 y축의 변수로하며, 거리를 기반으로하는 알고리즘 특성에 따라 R 코드의 scale함수를 이용해 단위를 표준화하였다. 그 결과 그래프의 x축은 악성행위 유형별 응답패킷의 길이로, y축은 악성행위 유형에 따른 *.php 패킷의 개수로 나타낼 수 있다.

악성트래픽 분석단계 결과는 Fig.8.과 같다. 결과적으로, 3개의 악성행위 유형별 군집화가 이루는 것을 볼 수 있다.

4.2 격리환경 구축

클라우드 환경은 레드햇 기반의 Openstack Platform 7.0을 적용하였다. 오픈스택이 지원하는 네트워크 기능을 통해 네트워크 주소 등 네트워크 정보를 설정하여 외부 네트워크를 구축하였다. 또한, 격리 네트워크 구축 시 외부 네트워크와 다른 서브넷을 부여함으로써 외부와 통신이 불가하도록 하였으며, 각 네트워크에 가상머신 인스턴스로 Windows 7을 설치하였다.

이에 따른 격리 네트워크 구축 결과는 Fig.9.와 같으며 이를 통해 격리 네트워크의 인스턴스는 외부와 통신이 차단된 상태임을 알 수 있다. 이때, 네트워크 토폴로지에서 외부 네트워크는 private 네트워크로 표시되며, 격리 네트워크는 isolation 네트워크로 표시된다.

격리환경 구축에는 볼륨 이동 방식을 사용하였다. 외부 네트워크의 인스턴스에 악성트래픽이 발생하면 인스턴스에 연결된 볼륨을 해제하여 해당 볼륨을 격리 네트워크 인스턴스에 연결한다. 볼륨은 인스턴스가 설치되는 저장 공간으로 인스턴스가 구동되는 과

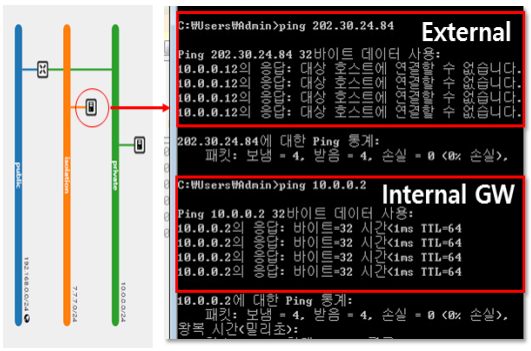

Fig. 9. Results of isolation network

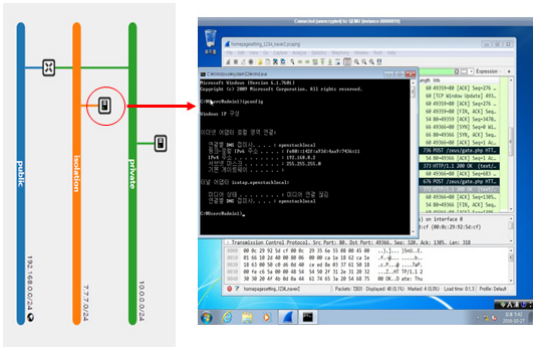

Fig. 10. Results of isolation-environment

정에서 메모리정보가 포함된다. 따라서, 볼륨 이동을 통해 인스턴스 복사가 가능하다. 격리환경을 구축한 결과는 Fig.10.과 같으며, 외부 네트워크의 인스턴스가 격리 네트워크의 인스턴스로 복사된 것을 알 수 있다.

4.3 악성행위 발현

악성행위 발현단계에서 악성트래픽 핸들링이 가능하기 위해 OvS 컨트랙 기능을 정적 분석한 결과는 Fig.11.과 같다. 이는 OvS 컨트랙 동작 흐름에 따른 함수 호출 관계 및 수행절차를 나타낸다.

이 과정에서 각 함수들은 네트워크 정보, 컨넥션 정보 등을 가리키는 구조체 포인터를 입력 파라미터로 사용한다. 이 중 *skb를 구조체 포인터로 갖는 sk_buff는 트래픽의 데이터 길이, 헤더 길이 등의 넷필터에 필요한 정보를 포함한다. 또한, *key를 구조체 포인터로 갖는 sw_flow_key는 컨트랙을 위한 이더넷 근원지 및 목적지 주소, 커넥션 상태와 같은 컨트랙 필드 등을 포함하며, *info를 구조체 포인터로 갖는 OVS_conntrack_info는 OvS 컨트랙 정보를 포함한다.

action.c의 do_execute_actions() 함수는 *skb가 가리키는 네트워크 정보를 기반으로 전송되는 트래픽에 대한 action 목록을 실행한다. 이때, OVS_ACTION_ATTR_CT 플래그를 통해 컨트랙 action을 정의하며 OVS_ACTION_ATTR_CT인 경우 ovs_ct_execute() 함수를 호출한다.

conntrack.c의 ovs_ct_execute() 함수를 통해 실제 action을 실행하기 위한 동작들이 구현된다. 이때 입력 파라미터인 *info가 커넥션 성립을 의미하는 commit을 가리키면, ovs_ct_commit()함수를 호출한다. 이 과정에서 넷필터 및 컨트랙을 위해 트래픽의 네트워크 정보, 커넥션 정보 등을 포함하는 *skb, *key 등이 입력 파라미터로 사용된다.

위 과정에서 호출된 ovs_ct_commit() 함수에서 커넥션 검색 및 커넥션 정의가 이루어진다. __ovs_ct_lookup() 함수를 통해 커넥션 상태를 검색하여 상태에 적합한 action 정보를 갱신한다. 커넥션 상태가 정의된 트래픽인 경우 nf_conntrack_confirm() 함수를 호출하여 action 동작을 전송함으로써 트래픽을 처리한다.

nf_conntrack_confirm() 함수의 경우 *skb가 가리키는 커넥션 정보를 확인한다. 이때, 커넥션이 확인되지 않으면 *skb 정보를 통해 커넥션을 정의하고 커넥션이 정의된 트래픽에 대해 해당하는 NF_ACCEPT, NF_DROP 등의 넷필터 정책을 수행한다. 반면, 트래픽의 커넥션이 확인되어 NF_ACCEPT가 설정된 경우 nf_ct_deliver_cached_events() 함수를 호출하여 해당 action을 트래픽을 통해 전달한다.

결과적으로, 악성트래픽의 커넥션 상태를 확인하고, 구조체 포인터가 가리키는 네트워크 주소 등을 변경함으로써 악성트래픽을 핸들링하여 재전송 공격을 할 수 있다. 이에 따라, 도출한 악성트래픽 재전송 공격을 위한 OvS 명령 단계는 Fig.12.와 같으며 동작 단계의 설명은 아래와 같다. 따라서, 해당 명령의 과정을 통해 악성트래픽의 재전송 공격이 가능하다.

OvS는 패킷처리를 위해 match와 action 항목을 필요로 한다. match는 OvS를 통해 들어오는 패킷의 플로우 정보를 매칭하기 위한 항목이다. match를 위한 변수 중 in_port는 OvS 브릿지에 연결되는 포트이고, state는 트래픽의 네트워크 커

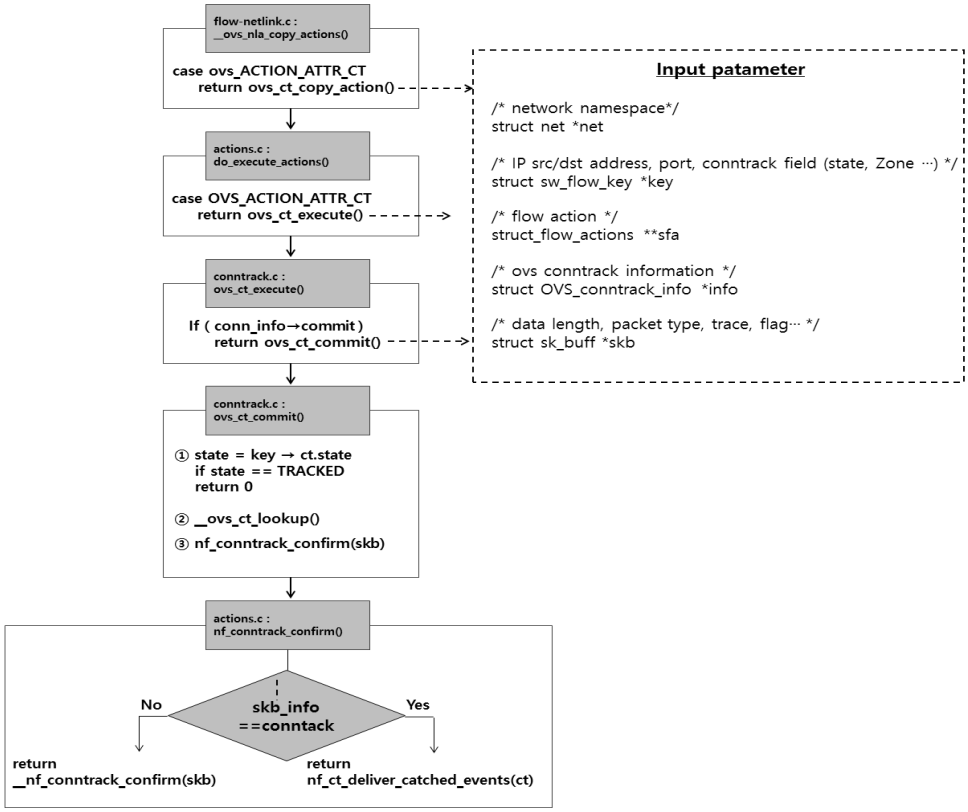

Fig. 11. Static analysis result of OvS conntrack function

넥션 상태를 정의한다. 반면, action은 OvS를 통해 나가는 패킷의 처리 행위를 정의하는 항목이다.

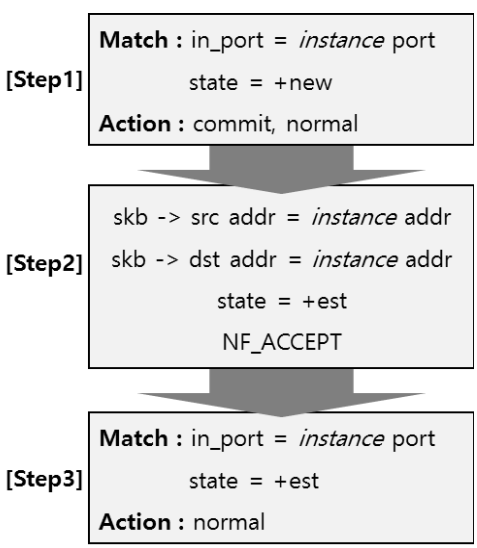

Fig. 12. OvS command step for replay attacks

Step1. in_port=인스턴스 port, state=+new에 match되는 악성트래픽에 commit 및 normal action 수행

이때, 인스턴스는 악성트래픽 분석을 위한 가상머신 인스턴스를 말한다. 인스턴스 포트로 들어오는 새 트래픽은 OvS 컨트랙 모듈에 이전 커넥션 정보가 없는 +new 상태로 매칭된다. 해당 정보로 매칭되는 악성 트래픽에 커넥션 성립 보장을 위한 commit과 기존의 트래픽 처리 프로세스 진행을 위한 normal action을 부여한다. 이에 따라 커넥션이 성립된 악성트래픽에 대한 커넥션 정보, 네트워크 정보 등은 컨트랙 모듈에 저장된다.

Step2. 악성트래픽 근원/목적 주소 및 state 변경 후 AF_ACCEPT 정책 부여

*skb가 가리키는 악성트래픽의 근원지 및 목적지 주소를 해당 인스턴스 주소로 변경함으로써 분석환경 내에서만 악성행위를 일으킬 필요가 있다. 이는 4.3절에서 분석한 ovs_ct_commit() 함수에서 커넥션

정보를 비교하여 커넥션 성립을 결정하는 과정 이후에 이루어져야한다.

주소가 변경된 악성트래픽의 커넥션 상태를 +est로 변경하여 OvS 컨트랙 모듈에 이미 존재하는 커넥션임을 알린다. 주소 변경으로 인해 기존에 저장되어있던 정보와 매칭되는 정보가 없기 때문에 임의로 변경해줄 필요가 있다. 이 또한 ovs_ct_commit() 함수에서 state 변수에 *key가 가리키는 커넥션 상태를 est로 변경함으로써 가능하다.

결과적으로 주소 및 커넥션 상태가 변경된 악성트래픽에 넷필터의 NF_ACCEPT 정책을 부여함으로써 트래픽 흐름을 허가한다. 이는 4.3절에서 분석한 내용을 바탕으로 넷필터 정책을 통해 action 동작을 정의하는 nf_conntrack_confirm() 함수에서 구현될 수 있다.

Step3. in_port=인스턴스 port, state=+est으로 match 되는 악성트래픽에 normal action 수행

2단계에서 변경한 +est 커넥션 상태를 갖는 악성트래픽에 normal action을 수행함으로써, 앞서 수행한 단계들로부터 변경된 정보로 핸들링 가능하다.

V. 기존 연구와의 비교

5.1 악성트래픽 분석 및 탐지

Laheeb 등[11]은 역공학을 이용한 Zeus 봇넷 분석 및 탐지 시스템을 제안하였다. Zeus 봇넷의 exe 파일을 ollydbg 역공학 도구를 이용해 CPU, 메모리, 로그, 설치 경로 등을 분석한 후 분석 데이터를 생성한다. 이를 기반으로 Zeus 봇넷을 탐지하며, 이후 exe 파일과 생성한 분석 데이터를 제거한다. 하지만, 해당 시스템은 탐지 이후 분석 데이터를 제거하기 때문에 공격이 발생할 때마다 역공학 분석이 필요하게 된다. 따라서 매번 역공학 분석이 필요함에 따른 시간이 소모되며, 추후 발생될 수 있는 변종 악성트래픽의 탐지가 어렵다.

반면, 제안하는 악성트래픽 탐지 방안은 봇과 봇넷간의 통신패턴을 분석하고, 클러스터링을 통해 악성트래픽을 분류된 공격 유형에 군집화시킴으로써 탐지하는 방식이다. 따라서 공격이 발생할 때마다 해당 파일을 정적으로 분석하여 탐지하는 방식에 비해 비교적 시간이 적게 소모된다. 또한, 통신패턴 분석을 통해 공격 유형별 군집화한 정보는 클러스터링 알고리즘에 지속적으로 업데이트됨으로써 다른 변종 악성트래픽의 탐지가 가능하다. 악성트래픽 분석 및 탐지 방안에 대한 기존 연구와 비교한 내용은 Table 2.와 같다.

악성트래픽의 암호화된 페이로드를 복호화하여 분석 및 탐지하는 방안들 또한 제시되고 있지만[6], 악성트래픽은 역공학을 방지하기 위해 난독화된 형태로 배포되기 때문에 복호화가 가능하더라도 난독화된 페이로드 분석에 추가적인 연구가 필요하다[12]. 따라서, 복호화 및 역공학과 같은 악성트래픽의 코드를 분석하는 방안은 많은 시간을 요구하며, 직접적인 분석이 어렵다. 그러므로, 이러한 난독화는 악성트래픽 분석 및 탐지에 있어 한계점으로 작용될 수 있다.

Table 2. Comparison with existing system (malicious traffic detection and analysis)

	Laheeb[11]	Proposed System
Method	Reverse engineering analysis of malicious traffic exe. files	Clustering through communication pattern analysis by attack type
Analysis data	Remove after detection	update after detection
Relative Time Cost	High	Low
Variant Detection	×	○

5.2 악성트래픽 핸들링

Mariano Graziano 등[5]은 격리된 분석환경에서 악성트래픽이 악성행위를 지속적으로 일으키게하기 위해 프록시 기술을 이용한다. 즉, 분석환경에 전송된 악성트래픽이 외부와 통신할 수 없도록 프록시 기술을 이용해 리다이렉션 시킴으로써 분석환경 내에서만 악성행위를 일으키도록 한다.

반면, 제안하는 시스템은 OvS 컨트랙 기능을 이용해 악성트래픽을 커널단에서 핸들링한다. 이는 분석환경에 악성트래픽이 전송될 경우 커널단의 OvS 컨트랙 모듈에서 악성트래픽을 핸들링하여 분석환경에서만 악성행위를 일으키도록하는 방안이다.

프록시와 같은 기존의 트래픽 핸들링을 위한 패킷 스위칭 등의 기능은 L7상의 컨트롤러(controller)

를 이용하여 가능하다. 이를 위해, L6을 비롯한 하위 계층의 네트워크 자원을 사용해야 한다. 반면, 본 논문에서 제안하는 악성트래픽 핸들링 방안은 OvS 커널에서 구현을 필요로하는 방안으로 L3 계층의 자원을 필요로 한다. 이는 트래픽 핸들링 과정에서 안전한 자원사용에 있어 기존의 방식보다 보안적 측면에서 고려해야할 사항이 적다. 이에 따른 악성트래픽 핸들링 방안에 대한 기존 연구와 비교한 결과는 Table 3.과 같다.

Table 3. Comparison with existing system (malicious traffic handling)

	Mariano Graziano[5]	Proposed System
Method	Traffic redirection through proxy	Traffic handling at the kernel level using OvS conntrack
Required Resources	Resources below L7	Resources below L3
Pros and Cons	Many resources to consider in terms of security	Few resources to consider in terms of security

VI. 결 론

본 논문은 안전한 클라우드 환경 제공을 목적으로 클라우드 환경에서의 악성트래픽 동적 분석 시스템을 제안하였다. 이 과정에서 k-means 클러스터링을 통한 악성트래픽 판별 및 분석 방안, 격리환경 구축을 통한 악성트래픽 분석환경 구축 방안 및 격리환경에서 OvS 컨트랙을 통한 악성트래픽 재전송 공격 방안을 제안하였다.

특히, 본 논문에서 제안하는 악성트래픽 재전송 공격 방안은 OvS 컨트랙을 통해 L3상의 자원을 이용하는 방안이다. 이에 따라, 기존의 트래픽 핸들링 방식보다 자원 사용에 있어 고려해야할 사항을 최소화함으로써 비교적 안전하고 효율적인 동적 분석이 가능하다.

추후 연구로는 Zeus 봇넷을 비롯한 다양한 악성트래픽에 적용될 수 있는 범용적인 클러스터링 알고리즘을 개발하고자 한다. 더불어, 분석 및 연구된 내용을 바탕으로 OvS 컨트랙을 이용한 악성트래픽 재전송 공격의 실증 연구를 진행할 예정이다.

본 논문에서 제안한 클라우드 환경에서의 동적 분석 시스템은 클러스터링, 악성트래픽 재전송 공격 등의 기법을 포함한다. 이러한 전 단계가 반복하면서 분석된 악성트래픽 정보는 지속적으로 업데이트될 수 있다. 이를 통해 향후 클라우드 환경에서 발생할 수 있는 신·변종 악성트래픽의 탐지 및 대응에 활용될 것으로 기대된다.

References

[1] KISA, "2016 internet and information security top ten issues outlook", Dec. 2015

[2] Shin Yeong-Sang, "Hypervisor-based virtualization security technology trend in the cloud environment", Jul. 2014

[3] DDaily, 2010, http://www.ddaily.co.kr/news/article.html?no=67684

[4] AhnLab, 2016, http://www.ahnlab.com/kr/site/product/cloudMss.do

[5] Mariano Graziano, Corrado Leita, Davide Balzarotti, "Towards Network Containment in Malware Analysis Systems", ACSAC '12 Proceedings of the 28th Annual Computer Security Applications Conference, pp.339-348, Dec. 2012

[6] H. Binsalleeh, T.Ormerod, A. Boukhtouta et al., "On the Analysis of the Zeus Botnet Crimeware Toolkit", Eighth Annual International Conference on Privacy, Security and Trust, Aug. 2010

[7] Baek Seung-Hun, "Open vSwitch packet processing structure", PIOLINK, Mar. 2015

[8] http://openvswitch.org/releases/NEWS-2.5.0

[9] Justin Pettit, Thomas Graf, "Stateful Connection Tracking & Stateful NAT", Nov. 2014

[10] A-Young Son, Eui-Nam Huh, "A Study on migration for QoS in cloud computing", Proceedings of Symposium of the Korean Institute of communications and Information Sciences, Jan. 2016

〔11〕 Laheeb Mohammed Ibrahim, Karam H. Thanon, "Analysis and Detection of the Zeus Botnet Crimeware", International Journal of Computer Science and Information Security, Vol. 13, No. 9, Sep. 2015
〔12〕 Babak Yadegari, Brian Johannesmeyer, Benjamin Whitely et al., "A Generic Approach to Automatic Deobfuscation of Executable Code" 2015 IEEE Symposium on Security and Privacy, May. 2015

〈저자소개〉

이 은 지 (Eun-ji Lee) 학생회원
2016년 2월: 공주대학교 정보통신공학과 학사
2016년 3월~현재: 아주대학교 컴퓨터공학과 석사과정
〈관심분야〉 제어시스템 보안, 클라우드 컴퓨팅 보안, 암호프로토콜, 사물인터넷 보안

곽 진 (Jin Kwak) 종신회원
2000년 8월: 성균관대학교 학사
2003년 2월: 성균관대학교 석사
2006년 2월: 성균관대학교 박사
2006년 4월~2006년 11월: 일본 큐슈대학교 방문연구원
2006년 8월~2006년 11월: 일본 큐슈시스템정보기술연구소 특별연구원
2006년 11월~2007년 2월: 정보통신부 정보보호기획단 개인정보보호팀 통신사무관
2007년 3월~2015년 2월: 순천향대학교 정보보호학과 교수
2008년 1월~현재: 한국정보보호학회 상임이사
2011년 1월~현재: 한국정보처리학회 이사
2015년 3월~현재: 아주대학교 사이버보안학과 교수
〈관심분야〉 자동차 보안, 암호프로토콜, 응용시스템보안, 클라우드 컴퓨팅 보안, 개인정보보호, 정보보호제품평가