ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

클라우드 환경에서 제우스 Botnet 공격 유형 분석을 위한 클러스터링 방안 연구

A Study on the Clustering method for Analysis of Zeus Botnet Attack Types in the Cloud Environment

배 원 일[1] 최 석 준[1] 김 성 진[2] 김 형 천[2] 곽 진[3*]
Won-il Bae Suk-June Choi Seong-Jin Kim Hyeong-Cheon Kim Jin Kwak

요 약

최근 클라우드 컴퓨팅 기술의 발전으로 인해 다양한 분야에서 클라우드 컴퓨팅 기술이 활용되고 있다. 클라우드 서비스의 수요가 증가하는 반면에 클라우드 환경에서의 보안 위협은 증가하고 있으며 특히, 악성코드에 의한 공격을 통해 클라우드 환경 내 상호 연결되어 있는 호스트들이 감염 전파될 경우 다른 호스트의 리소스에도 영향을 끼쳐 개인정보 및 데이터의 삭제 등의 보안위협이 확산될 수 있다. 따라서 이러한 보안 위협에 대응하기 위한 악성코드 분석 연구가 활발히 진행되고 있다. 이에 따라, 본 논문은 클라우드 환경에서 발생하는 악성코드 분석을 위해 k-means 클러스터링 알고리즘을 이용한 제우스 봇넷의 공격 유형별 군집화 방안을 제안한다. 이는 클라우드 환경 내 발생되는 제우스 봇넷에 대하여 악성행위를 유형별로 군집화 함으로써 악성 유무를 판별할 수 있으며, 추후 클라우드 환경에서 발생할 수 있는 새로운 유형의 제우스 봇넷 공격 대응을 목표로 한다.

☞ 주제어 : 클라우드 컴퓨팅, K-means 클러스터링, 제우스 봇넷, 오픈스택

ABSTRACT

Recently, developments in the various fields of cloud computing technology has been utilized. Whereas the demand for cloud computing services is increasing, security threats are also increasing in the cloud computing environments. Especially, in case when the hosts interconnected in the cloud environments are infected and propagated through the attacks by malware. It can have an effect on the resource of other hosts and other security threats such as personal information can be spreaded and data deletion. Therefore, the study of malware analysis to respond these security threats has been proceeded actively. This paper proposes a type of attack clustering method of Zeus botnet using the k-means clustering algorithm for malware analysis that occurs in the cloud environments. By clustering the malicious activity by a type of the Zeus botnet occurred in the cloud environments. it is possible to determine whether it is a malware or not. In the future, it sets a goal of responding to an attack of the new type of Zeus botnet that may occur in the cloud environments.

☞ keyword : Cloud computing, K-means clustering, Zeus botnet, Openstack

1. 서 론

최근 클라우드 컴퓨팅(Cloud computing)기술의 발전으로 인해 다양한 클라우드 기반 플랫폼이 활성화되고 있는 반면에 클라우드 환경에 보안 위협은 증가하고 있는 추세이다. 특히 클라우드 컴퓨팅 환경의 특성상 다수의 가상머신(Virtual Machine)을 사용자에게 제공하기 때문에 악성코드로 인한 보안 위협이 크게 대두되고 있는 실정이다[1].

대표적 악성코드인 제우스 봇넷(Zeus botnet)은 국외에서 다양한 금융 피해 사례가 발생하고 있으며 클라우드 컴퓨팅 환경에서 인가되지 않은 디바이스를 이용한 접근으로 클라우드 환경 내에서 제우스 봇넷과 같은 악성코드에 의한 보안 위험성이 높아지고 있다.

제우스 봇넷은 감염된 봇과 통신하기 위해 설정파일을 요청 및 응답 과정을 수행한다. 하지만 이러한 설정파일은 공격자가 임의로 이름을 지정을 해줄 수 있기 때문에

1 Department of Computer Engineering, Ajou University, Suwon, 16499, Korea.
2 National Security Research institute, Daejeon, 34044, Korea
3 Department of Cyber Security, Ajou University, Suwon, 16499, Korea.
* Corresponding author (jkwak.security@gmail.com)
[Received 27 October 2016, Reviewed 29 October 2016, Accepted 22 November 2016]

탐지가 어렵다. 또한 클라우드 환경 기반의 서비스를 이용하는 회사를 대상으로 로그인 계정을 노리는 변종 제우스 봇넷이 발견되었으며[2] 이에 따라 클라우드 컴퓨팅 환경에서도 제우스 봇넷과 같이 점점 지능적으로 변해가는 악성코드들에 대한 대책이 필요한 상황이다.

따라서 클라우드 컴퓨팅 환경에서 제우스 봇넷과 같은 악성코드 탐지에 대한 연구가 활발히 진행되고 있다[3].

본 논문에서는 클라우드 컴퓨팅 환경을 구축하고 제우스 봇넷의 악성 트래픽을 발현하기 위해 제우스 봇넷을 구축하였다. 또한 제우스 봇넷의 악성 패킷을 분석하고 알고리즘을 통해 추출되어진 데이터 셋을 활용하여 클러스터링(Clustering)을 수행함으로써 제우스 봇넷의 공격 명령에 대한 유형별 군집 결과를 도출한다.

2장에서는 제우스 봇넷과 제우스 봇넷의 네트워크 통신 및 다양한 공격 유형들 및 k-means 클러스터링 기법에 대해 분석하고, 3장에서는 제안 기법에 대한 시나리오와 공격 유형별 분석 기법 및 클러스터링 기법에 대해 제안하며, 4장에서는 제안하는 기법을 이용한 실험 결과를 도출한 후, 5장에서 실험 결과에 대한 오탐을 분석하고, 6장에서 본 논문에서 제안한 기법과 기존 연구를 비교한다. 마지막 7장에서 결론으로 끝을 맺는다.

2. 관련 연구

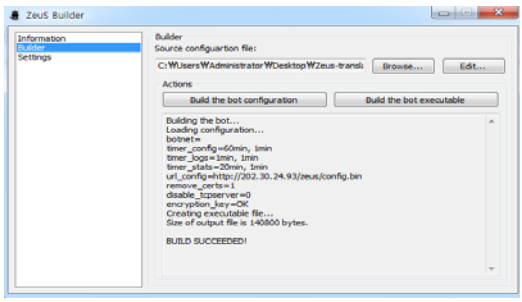
2.1 제우스 봇넷

2007년 러시아에서 처음으로 개발된 것으로 추정되는 제우스 봇넷은 공격자에 의해 생성된 트로이 목마의 형태를 가지고 있으며 감염률이 가장 높은 대표적인 악성코드이다. 제우스 봇넷은 모바일 환경 및 클라우드 환경에서 동작할 수 있으며 지능화되고 있기 때문에 다양한 IT 인프라 시스템이 대상이 된다[4].

제우스 봇넷은 다양한 경로로 전파될 수 있으며 스팸메일, 소셜 네트워크 및 악성 스크립트 등을 통해 쉽게 전파될 수 있기 때문에 전 세계적으로 제우스 봇넷의 C&C 서버가 분포되어 있으며 특히 미국, 유럽 지역을 중심으로 해외 금융 피해 사례가 발생하였고 2011년 국내 금융기관을 겨냥한 제우스 봇넷 유출 소스가 발견되었다.

또한 제우스 봇넷은 사용자의 컴퓨터를 감염시킬 수 있는 봇과 이를 생성하는 봇넷으로 구성되어 있으며 (그림 1)과 같은 봇넷의 빌더(Builder)를 이용하여 감염된 봇에게 다양한 유형의 공격 명령을 수행할 수 있다.

제우스 빌더로 생성된 봇은 감염된 사용자의 컴퓨터의



(그림 1) 제우스 봇넷 빌더
(Figure 1) Zeus botnet builder

봇넷 이름, 봇의 ID, 봇의 버전 등 봇의 정보와 운영체제의 버전 및 언어, 지역 및 시간, IP 주소 등과 같은 시스템 정보뿐만 아니라 사용자가 웹페이지에 입력한 모든 값들을 저장하여 C&C 서버로 전달하고 화면 캡쳐 기능을 통해 사용자의 거래 정보 및 개인정보를 수집할 수 있다[5].

2.1.1 제우스 봇넷의 네트워크 통신

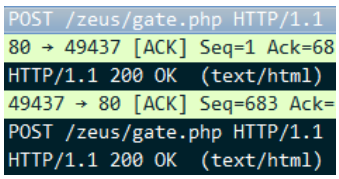
본 절에서는 봇과 봇넷의 네트워크 통신에 대한 내용을 다룬다. 제우스 봇넷은 봇과 봇넷 간의 HTTP 기반 통신을 수행하며 모든 통신은 RC4 암호화 가 되어있다. 봇넷에 의해 생성된 봇이 사용자의 컴퓨터를 감염시키면 (그림 2)와 같이 봇은 봇넷에게 환경설정 파일인 config.bin*파일을 GET 방식으로 요청하고 봇넷은 암호화된 config.bin을 전송함과 동시에 200 OK라는 응답 메시지를 전송한다.

```
GET /zeus/config.bin HTTP/1.1
[TCP segment of a reassembled PDU]
49432 → 80 [ACK] Seq=276 Ack=1461 Win=65536 Len=0
[TCP segment of a reassembled PDU]
49432 → 80 [ACK] Seq=276 Ack=2921 Win=65536 Len=0
[TCP segment of a reassembled PDU]
49432 → 80 [ACK] Seq=276 Ack=5841 Win=65536 Len=0
[TCP segment of a reassembled PDU]
49432 → 80 [ACK] Seq=276 Ack=8761 Win=65536 Len=0
[TCP segment of a reassembled PDU]
49432 → 80 [ACK] Seq=276 Ack=17521 Win=65536 Len=0
HTTP/1.1 200 OK  (application/octet-stream)
```

(그림 2) Config.bin 파일의 요청과 응답 패킷
(Figure 2) Request and response of config.bin

봇넷과 봇이 환경설정 파일에 대한 요청과 응답을 완료하면 (그림 3)과 같이 감염된 사용자 컴퓨터의 봇은 정보를 유출하고 봇의 상태를 업데이트하는 gate.php**를 두

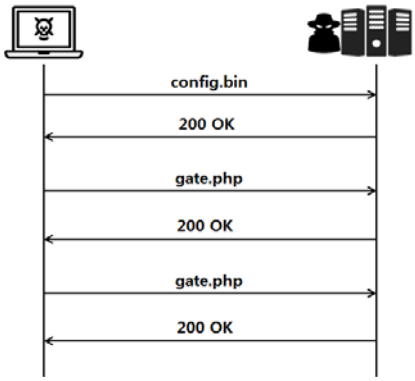
* 일반화된 *.bin 파일이며 환경설정에 따라 이름이 바뀔 수 있음



(그림 3) gate.php 파일의 요청과 응답
(Figure 3) Request and response of gate.php

번 전송한다. 봇넷은 gate.php 파일을 수신한 후, 200 OK 라는 응답 메시지를 전송하며 초기 통신의 gate.php파일은 봇의 쿠키 정보를 유출한다[6].

초기 통신 수행을 마친 이후에 봇넷의 공격 명령에 따라 사용자 컴퓨터의 정보를 유출하는 gate.php를 송수신 과정을 재 수행한다. 봇넷과 봇의 초기 통신 과정은 (그림 4)와 같다[6].



(그림 4) 봇넷과 봇의 초기 통신 과정
(Figure 4) Initial communication pattern

2.1.2 제우스 봇넷의 공격 유형

제우스 봇넷은 (표 1)과 같이 다양한 공격 명령 외에도 봇의 상태를 업데이트하고 봇의 이름 수정 및 사용자의 컴퓨터에서 봇을 삭제하는 기능 등을 수행할 수 있다[7].

(표 1) 제우스 봇넷의 공격 명령 종류
(Table 1) Type of Zeus botnet attack scripts

No	Type	Description
1	user_flashplayer_get	gets the Flash player data form a user's system
2	user_ftpclients_get	steals passwords from FlashFXP,total_commander,ws_ftp,fileZilla, FAR2, winscp, ftp_commander, coreFTP and smartftp
3	user_homepage_set	set the browser's home page to desired URL
4	user_url_unblock	restore access to an attacker-desired URL
5	user_url_block	disable access to an attacker-desired URL
6	user_certs_remove	removes certificate
7	user_certs_get	steal digital certificates
8	user_cookies_remove	delete browser cookies
9	user_cookies_get	upload cookies
10	user_execute	download and execute a file
11	user_logoff	log off user
12	bot_bc_remove	Removes a back door connection
13	bot_bc_add	initiate back door by back-connecting to a server and allow arbitrary command execution via the command shell
14	bot_update	download and update bot config (sets in registry as well), download and execute new bot installer
15	bot_uninstall	remove bot altogether
16	os_reboot	reboot the computers
17	os_shutdown	shut down the computer

** 일반화된 *.php 파일이며 환경설정에 따라 이름이 바뀔 수 있음

2.2 K-means 클러스터링

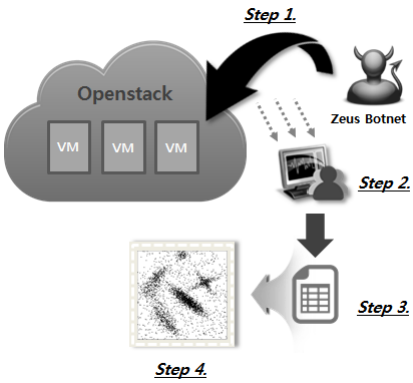
K-means 클러스터링이란, 각 데이터 셋이 주어 졌을 때, 주어진 데이터 셋의 거리의 차를 기반으로 가장 가까운 거리에 있는 데이터를 군집화 하는 Centroid model 중 하나이다. Centroid란 각 클러스터들의 중심 값을 말한다. 이는 클러스터들인 k개의 중심 값인 mean값에 따라 클러스터들의 Centroid값이 결정되며, 기준에 따라 데이터 들이 군집된다[8]. 현재 이 알고리즘은, 기계학습에서도 무수히 많이 활용 되고 있으며, 추가적인 데이터에 따라 기계적인 학습이 가능한 알고리즘이다. 본 논문에서는 R코드를 활용하여 k-means클러스터링을 진행한다.

R코드는 통계 분석과 그래프를 그리는데 효율적인 분석이 가능한 언어이며 이를 사용하여, 저장된 데이터 셋에 따라 k-means 클러스터링을 진행하게 된다[9].

3. 제안 기법

3.1 제안 기법 시나리오

본 논문에서 제안하는 클라우드 환경에서 제우스 봇넷 공격 유형 분석의 시나리오는 총 4단계로 이루어졌으며 (그림 5)와 같다.



(그림 5) 제안하는 시나리오
(Figure 5) Proposal scenario

Step 1.

클라우드 환경에서 제우스 봇넷 발현을 위하여 오픈소스 클라우드 환경인 오픈스택(Openstack)으로 구축하고 클라우드 환경의 외부 가상머신에 봇넷을 설치한다.

Step 2.

클라우드 환경 외부의 봇넷에서 생성된 봇을 클라우드 환경의 인스턴스에 설치한 후, 감염시켜 봇넷과 봇의 통신을 가능하게 함으로써 제우스 봇넷의 트래픽을 발생시킨다.

Step 3.

봇넷의 컨트롤 패널(Control panel)에서 스크립트 커맨드(Script command)를 이용한 공격 명령을 내리게 되면 클라우드 환경의 감염된 인스턴스는 공격 명령에 대한 트래픽을 발생시키게 되고 이러한 트래픽들을 수집하여 데이터 셋(Data set)을 생성한다.

Step 4.

제안하는 공격 유형별 분석 알고리즘을 이용해서 Step 3.에서 생성된 데이터 셋의 공격 명령에 대한 트래픽을 분석한다. 이에 대한 결과를 클러스터링하여 제우스 봇넷의 공격 유형을 판별한다.

3.2 제안하는 공격 유형별 분석 기법

제우스 봇넷의 공격 유형은 2.2절과 같이 여러 공격 유형이 존재하며 본 논문에서 제우스 봇넷의 공격 유형별 분석을 위하여 스크립트 커맨드를 이용한 3가지 공격을 수행하였다. 수행한 공격 명령은 (표 2)와 같이 감염된 봇을 종료시키는 os_shutdown와 explorer의 홈페이지를 봇넷에서 설정한 url로 지정하는 user_homepage_set 및 감염된 봇의 쿠키 정보를 탈취하는 user_cookies_set과 같다.

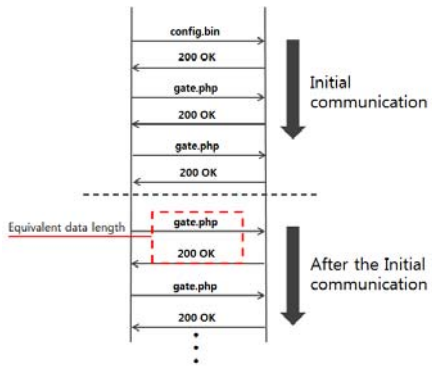
(표 2) 선정한 3가지 공격 유형
(Table 2) Three kinds of Zeus Attack

No.	Type	Description
1	os_shutdown	Shutdown computer
2	user_homepage_set	Set homepage [url] for Internet Explorer
3	user_cookies_get	Upload cookies

본 논문에서는 제우스 봇넷에 대한 요청과 응답 쌍을 분석하였으며 제우스 악성 코드의 공격 유형에 따라 2가지 유형별 특징을 도출한다.

3.2.1 첫 번째 특징

제우스 봇넷의 공격 유형의 첫 번째 방법은 감염된 봇



(그림 6) 초기 통신 이후 gate.php 파일의 응답 패킷 데이터 길이

(Figure 6) The length of the equivalent response packet data of gate.php file after initial communication

의 정보 유출을 가능하게 하는 php 파일의 응답 데이터의 길이이다. 제우스 봇넷과 봇 간에 공격 명령을 수행 시 감염된 봇은 config.bin 파일에 작성된 gate.php 파일을 봇넷에게 요청하며 봇넷은 200 OK라는 응답으로 초기 통신을 마친다. 초기 통신 이후 공격 명령에 따라서 gate.php파일에 대한 요청과 응답을 반복하게 되며 (그림 6)과 같이 초기 통신 직후 요청된 gate.php 파일에서 동일한 공격 명령에 따라 응답 패킷 데이터의 길이가 동일하다.

이를 통해 정보 유출을 가능하게 하는 gate.php파일에 대한 응답 데이터의 길이가 동일한 공격 유형은 같은 공격 유형이라고 간주할 수 있다.

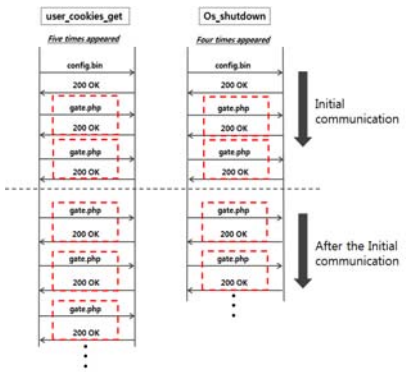
3.2.2 두 번째 특징

공격 유형을 판별하는 두 번째 방법은 봇넷과 봇 간의 초기 통신 이후, 공격 명령에 의해 발생하는 정보 유출 파일인 gate.php의 횟수이다.

C&C서버의 컨트롤 패널에서 스크립트 커맨드를 이용하여 user_cookies_get 명령과 os_shutdown 명령을 수행하게 되면 (그림 7)과 같이 초기 통신을 포함한 gate.php 파일의 횟수가 각각 5회, 4회로 다르게 나타남으로써 공격 명령에 대한 유형을 분석할 수 있다.

3.3 제안하는 클러스터링 모델

클러스터링 수행을 위하여 공격 유형별 데이터 셋을 필요로 한다. (그림 8)에서 제안하는 알고리즘을 이용하여



(그림 7) gate.php의 길이 비교

(Figure 7) Comparison of gate.php file frequency

(그림 8) 제안하는 흐름도

(Figure 8) Proposal flowchart

공격 명령에 대한 앞서 제안한 두 가지 특징을 추출함으로써 클러스터링에 필요한 데이터 셋을 생성할 수 있다.

Step 1. 악성 트래픽 탐지

분석하고자 하는 pcap파일을 입력하여 알고리즘에 적용할 경우 가장 먼저 봇넷과 봇간의 초기 통신에서 요청 및 응답하는 config.bin이 존재하는지 확인 한다[6]. config.bin은 제우스 봇넷이 발현되었을 때 초기 통신에서 발견되는 환경설정 파일이므로 이를 통하여 제우스 봇넷의 감염 유무를 확인할 수 있다. 만약 검색되지 않을 경우, 제우스

봇넷에게 감염되지 않음을 판단하고 패킷 패일의 분석을 중지한다.

Step 2. 공격 유형별 분석

초기 연결 시 발견되는 패킷인 config.bin이 존재하는 경우 해당 패킷이 어떤 유형의 공격 명령을 내리는 패킷인지 알고리즘을 통해 판단 한다. 해당 알고리즘은 총 3개의 유형을 검색하며, 각 유형에 해당하는 조건은 gate.php의 요청 및 응답 쌍의 개수인 *.php count와 해당 응답 패킷의 길이인 length값을 비교한다. 첫 번째 조건인 User_cookies_get의 유형에 해당 할 경우, 해당 데이터 셋을 추출하고, 해당하지 않을 경우에는 User_homepage_set, OS_Shutdown의 조건에 만족하는지 차례대로 검색 한다. 만약 모든 유형에 해당하지 않을 경우에는 Step 1.의 악성 트래픽을 분석하기 위한 pcap파일을 입력하는 부분으로 돌아가며, 이를 반복 수행 한다. 또한 해당 유형이 맞을 경우, 이에 대한 데이터 셋은 powerGraphData.csv로 출력을 한다.

Step 3. 클러스터링

Step 2.에서 분석을 통해 만들어진 데이터 셋은, 봇이 전송하는 gate.php에 대한 응답 패킷 데이터의 길이와 POST 방식으로 전송하는 gate.php쌍의 개수로 구성되며, k-means를 통해 클러스터링을 하게 된다. 이에 해당하는 k값은 각 공격 유형의 종류에 따라 선정할 수 있으며, 본 논문에서는 k의 값을 3으로 선정하였다. 이러한 공격 유형을 통해 클러스터링을 진행하며, 향후 제우스 봇넷의 공격에 대응하기 위한 방안으로 활용한다.

본 논문에서는 4장의 실험 결과를 통해 클러스터링에 대한 산출 결과 및 구체적인 내용을 설명 한다.

4. 실험 결과 및 분석

본 절에서는 클라우드 환경에서 제우스 봇넷의 공격 유형 분석 및 클러스터링 과정 및 결과를 설명한다.

4.1 실험 환경

본 논문의 실험 환경은 클라우드 환경을 구축하기 위하여 오픈스택 Kilo 버전으로 구축하고 감염된 봇 환경을 위한 가상머신을 생성한다. 또한 클라우드 환경의 외부 가상머신 환경은 VMware이며 VMware의 가상머신에 제우스 봇넷을 설치한다.

본 논문에 사용된 제우스 봇넷은 제우스 2.0.8.9 버전이며 봇넷과 봇 간의 패킷 수집을 위하여 Wireshark를 이용한다. 클러스터링을 위한 알고리즘은 파이썬 2.7버전으로 개발하며 Wireshark의 패킷을 활용하기 위하여 Pyshark 라이브러리를 임포트한다. 클러스터링은 R코드를 이용하여 구현하였으며 (표 3)과 같다.

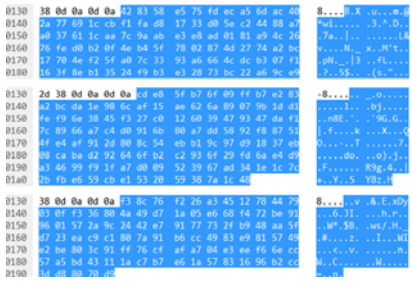
(표 3) 실험 환경
(Table 3) Experiment environments

No.	Function	Description
1	openstack	kilo
2	botnet environment	VMware
3	malware	zeus builder 2.0.8.9
4	packet capture	wireshark
5	algorithm	Python 2.7
6	clustering	R

4.2 실험 결과

클라우드 환경에서 공격 유형 분석 및 클러스터링을 위한 데이터 셋을 생성한다. 데이터 셋의 공격 유형은 3.2절과 같이 os_shutdown, user_homepage_set, user_cookies_get이며 초기 통신을 거친다고 가정하며 공격 유형 별로 10회의 공격을 시도하여 패킷을 수집한다.

수집한 데이터 셋의 분석 결과 초기 통신 직후 요청된 *.php 파일의 응답 패킷 데이터의 길이가 (그림 9)와 같이 공격 유형에 따라 다르게 나타난다.



(그림 9) 공격 유형에 따른 응답 패킷 데이터의 길이 비교
(Figure 9) length comparison of the response packet data according to the attack type

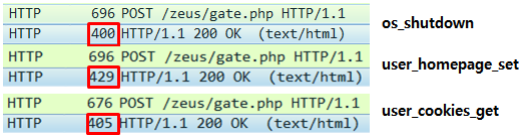
이에 대한 전체 응답 패킷 데이터의 길이를 비교하면 os_shutdown 공격 명령은 400bytes의 길이를 가지고 user_homepage_set의 공격 명령의 길이는 홈페이지 주소에 따

(표 4) 클러스터링을 위한 데이터셋
(Table 4) Dataset for k-means clustering

Attack	Type	1	2	3	4	5	6	7	8	9	10
shutdown	length	400	400	400	400	400	400	400	400	400	400
	*.php	4	4	4	4	4	4	4	4	4	4
homepage	length	431	433	431	431	432	433	432	431	431	432
	*.php	4	4	4	4	4	4	4	4	4	4
cookie	length	405	405	405	405	405	405	405	405	405	405
	*.php	5	5	5	5	5	5	5	5	5	5

라서 429~431bytes, user_cookies_get 공격 명령은 405bytes의 길이를 가지며 (표 10)과 같다.

따라서 공격명령에 따라 다른 응답 패킷 데이터 길이를 가진다.

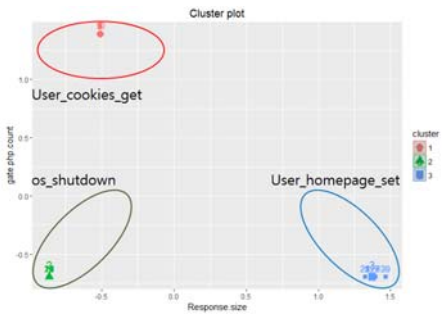


(그림 10) 세 가지 공격에 대한 응답 패킷 데이터의 길이 비교
(Figure 10) length comparison of the response packet data according to the each three attack type

봇넷과 봇 간의 초기 통신을 포함한 공격 명령에 의해 발생하는 정보 유출 파일인 *.php의 횟수를 분석하면 os_shutdown 공격 명령과 user_homepage_set의 공격 명령 모두 4회이며 user_cookies_get의 공격 명령은 5회의 *.php 파일이 발생된다.

따라서 본 논문에서 제안하는 제우스 봇넷의 os_shutdown, user_homepage_set, user_cookies_get 공격 명령에 대한 응답 패킷 데이터의 길이 및 gate.php의 횟수는 3.3절에서 제안한 알고리즘에 의해 (표 4)와 같이 데이터 셋을 추출하였다. 클러스터링 결과를 도출하기 위한 데이터 셋은 응답 패킷 데이터의 길이 Length, 데이터 유출과 관련된 *.php의 발생 횟수 쌍으로 이루어져 있고 공격 유형에 따라 10회씩, 총 30개의 데이터 셋을 추출 하였다.

표와 같이 Homepage set은 패킷의 길이가 고정적이지 않고 가변적인 모습을 보이고 있다. 이는 봇넷이 설정하고자 하는 홈페이지의 따라 패킷의 길이가 가변적으로 차이가 나기 때문이다. 이러한 데이터 셋에 따라, 클러스터링을 진행하게 된다. 본 논문에서 수행한 클러스터링 결과는 (그림 11)과 같다.



(그림 11) 클러스터링 결과
(Figure 11) Result of clustering

(그림 11)과 같이 적용된 k-means 알고리즘의 k값에 따라 클러스터의 개수를 지정할 수 있으며, 생성된 클러스터는 들어오는 데이터에 따라, 거리의 평균치를 산출하여 기준이 결정된다.

본 논문에서는 세 가지 공격 유형을 활용하여 데이터 셋의 공격 패킷을 클러스터링하기 때문에 k값을 3으로 선정하였다. 또한 이러한 k-means 알고리즘은 통계 분석이 가능한 R코드의 클러스터 라이브러리를 추가하여 적용을 하였다. 또한 k-means를 그래프로 표현하기 위해서 연산된 값을 가시화하기 위해 factoextra 라이브러리를 추가하여 fviz_cluster를 적용하였다[10]. 또한 해당 결과의 x축과 y축의 범위를 정규화 하기 위한 방법으로 scale함수를 이용해 정규화 하였다. 이에 따른 각 클러스터별 데이터에 분포된 평균값은 (표 5)와 같다.

(표 5) 클러스터 별 데이터 평균 값
(Table 5) Cluster value of summary

K	Response.size	gate.php.count
1	431.8	4
2	400	4
3	405	5

(표 5)와 같이 현재 클러스터링한 K의 그룹은 총 3개의 클러스터로 homepage_set, os_shutdown, user_cookies_get 순으로 구성되어 있다. 또한 현내 그룹별 산출된 값은 각 클러스터별 분포되어 있는 패킷의 길이와 gate.php쌍의 평균 값을 나타낸다. 이에 따라 현재 제우스 봇넷의 공격 유형별로 군집화 된 것을 볼 수 있다.

5. 오탐 분석

본 절에서는 제우스 봇넷의 공격 명령에 대한 유형별 클러스터링 기법에 대한 오탐을 분석한다.

본 논문에서 클러스터링을 하기 위해 사용하는 k-means 기법은 데이터 셋을 기반으로 하여 군집이 형성되며 각 데이터의 거리를 이용해 군집을 나타낸다. 하지만 추가되는 데이터에 대한 특이값이 나타나게 되는 경우 올바르지 않는 군집이 생길 수 있다.

(표 5)와 같이 현재 shutdown과 cookie의 경우 x축에 해당하는 응답 패킷 데이터의 길이는 차이가 나지 않지만, y축의 해당하는 값인 gate.php의 개수가 다르기 때문에 올바르게 클러스터링이 수행된다. 하지만 추가되는 데이터에 의해서, 특이 값에 해당하는 데이터가 들어올 경우, 올바르지 않은 군집이 생길 수 있으며 이는 (표 6)과 같다.

(표 5) 오탐된 클러스터별 데이터 평균 값
(Table 5) False positives Cluster value of summary

K	Response.size	gate.php.count
1	431.4286	4
2	435	4
3	402.5000	4.5

(표 6)의 특이값으로 인해 생성된 클러스터들의 평균 값은 user_homepage_set 공격 명령과 다른 공격 명령을 비교하였을 때, 공격자가 지정한 홈페이지 사이트에 따라 응답 패킷의 길이가 일정하지 않고 가변적이기 때문에 군집에 문제가 발생할 수 있다. 이로 인해 shutdown과 cookie의 클러스터가 하나로 묶이는 현상이 발생하게 된다.

이는 (표 4)와 같이 데이터 셋이 30개로 한정되어 있기 때문에 추가적인 데이터 생성으로 인한 특정 값으로 군집에 영향을 주며 이로 인한 오탐이 발생하게 되기 때문에 추가적인 데이터의 생성 및 재 수집을 진행해야 한다[11].

6. 기존 연구와의 비교

H.Binsalleeh 등[6]은 제우스 봇넷의 패킷을 분석하기 위해 공격 명령에 대한 응답 패킷 데이터의 페이로드를 분석한다. 페이로드는 RC4로 암호화 되어있으며 H.Binsalleeh 등[6]은 응답 패킷 데이터의 페이로드를 복호화하여 분석한다. 하지만 패킷의 페이로드를 복호화 하여 제우스의 공격 명령 패킷을 분석하는 접근 방법을 사용하게 될 경우 패킷에 대한 복호화 과정에 있어, 시간이 소모되며 복호화 한 후에도, 이러한 페이로드가 어떤 유형의 공격을 하는 지에 대한 분석 과정이 필요하다.

하지만 본 논문의 경우 이러한 공격 유형의 분석을 클러스터링 기법으로 진행하기 때문에, 여러 유형의 패킷에 따라 클러스터가 형성되고 이 클러스터를 통해 해당 패킷이 어떤 유형의 공격인지 판단할 수 있다. 따라서, 패킷의 페이로드를 복호화를 하여 공격 유형을 판단하는 방식의 경우 변종패킷이 들어올 때마다 재분석을 해야 하지만 본 논문에서는 신종 패킷이 들어올 경우 해당 공격 유형과 거리가 가까운 클러스터에 군집이 된다. 이는 변종 패킷이 들어올 경우에도 재분석을 할 필요가 없이, 해당 변종 패킷이 군집된 클러스터들 중 가장 가까운 거리에 속하게 된다. 이러한 방법론으로 인해 다른 변종패킷에 대해서도 유연하게 대응이 가능하다.

(표 6) 기존 연구와의 비교
(Table 6) Comparison with existing study

-	H.Binsalleeh et al.	Ours
Method	Payload decryption	Attack type pattern analysis
Result	Decrypted payload analysis	Attack type analysis through clustering
Pros and Cons	Perform reanalysis according to new malware packets	Cluster formation according to new malware packets

7. 결 론

클라우드 컴퓨팅 서비스는 IT 자원 및 다양한 데이터를 가상 스토리지에 저장하며, 해당 데이터는 민감한 정보가 포함되어 있다. 따라서 클라우드 컴퓨팅 환경은 공격자에 의해 알맞은 공격대상으로 가상머신이 악성코드에 의해 감염되었을 때, 개인 정보 유출 등 다양한 보안 위협이 발생할 수 있다[12].

본 논문에서는 클라우드 환경에서 대표적인 악성코드인 제우스 봇넷의 통신 패턴을 분석하고 제우스 봇넷의 공격 유형별 특징을 정의하였다.

이러한 공격 유형별 특징을 이용하여 3.3절에서 제안하는 알고리즘을 통해 제우스 봇넷 패킷에 대한 데이터 값을 추출하고 클러스터링 기법을 이용하여 제우스 봇넷의 공격 유형별로 군집화가 수행됨을 보였다. 또한 제우스 봇넷의 공격 명령은 내재된 특징이 공격 명령에 따라서 동일하기 때문에 기존 공격 명령의 유형을 수치화한다면 변형되거나 다른 공격 유형에 대한 대응이 가능할 것으로 기대된다.

향후에는 클러스터링 오탐에 영향을 미치는 특이값의 안정적인 결과에 대한 연구와 클라우드 컴퓨팅 환경에서 악성코드가 발현되었을 때, 악성코드가 발현된 가상머신을 격리시키고 외부 네트워크와 차단된 인스턴스 환경에서 재전송 공격을 통하여 악성코드가 발현되도록 유도하는 기술에 대해 연구를 진행할 예정이다[13].

참 고 문 헌(Reference)

[1] Michael R. Watson, Noor-ul-hassan Shirazi and Angelos K. Marnerides, “Malware Detection in Cloud Computing Infrastructures” IEEE Transactions on Dependable and Secure Computing. pp. 192-205, July. 2015. http://dx.doi.org/10.1109/TDSC.2015.2457918

[2] Marcos Colon, Dan Raywood, “http://www.scmagazineuk.com/new-variant-of-zeus-targets-logins-for-cloud-based-systems/article/236170/”, SC Magazine UK, April, 2012.

[3] Mark Graham, Adrian Winckles, “Botnet Detection within Cloud Service Provider Networks using Flow Protocols” INDIN 13th IEEE International Conference on Industrial Informatics, At Cambridge, July. 2015. http://dx.doi.org/10.1109/INDIN.2015.7281975

[4] N. Falliere, E. Chien “Zeus:King of the Bots Technical Report” Symantec, 2009. https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf

[5] Bill Buchanan, “Implementaion and Evaluation of a Botnet Analysis and Detection Methods in a Virtual Environment”, Edinburgh Naphier University, Aug 2012. http://www.iidi.napier.ac.uk/c/publications/publicationid/13373235

[6] H. Binsalleeh, T.Ormerod, “On the Analysis of the Zeus Botnet Crimeware Toolkit”, Eighth Annual International Conference on Privacy, Security and Trust, 2010. http://dx.doi.org/10.1109/PST.2010.5593240

[7] Philip Schwartz, “Setup and Analysis of Zeus Banking Trojan V2.0.8.9”, Volatility and LibVMI in a Virtualized lab, August, 2014. http://docplayer.net/12934657-Setup-and-analysis-of-zeus-banking-trojan-v-2-0-8-9-w-volatility-and-libvmi-in-a-virtualized-lab.html

[8] Jain, Anil K, ″Data clustering: 50 years beyond K-means,″ Pattern Recognition Letters 31, pp. 651-666, 2010. http://dx.doi.org/10.1016/j.patrec.2009.09.011

[9] J.A. Hartigan and M.A. Wong, ″Algorithm AS 136 : A K-Means Clustering Algorithm″, Journal of the Royal Statistical Society. Series C, pp. 100-108, 1979 http://dx.doi.org/10.2307/2346830

[10] Pamulaparty, Lavanya, CV Guru Rao, and M. Sreenivasa Rao. ″Cluster Analysis of Medical Research Data using R″, Global Journal of Computer Science and Technology. 2016.

[11] Khormali, Aminollah, and Jalil Addeh. ″A novel approach for recognition of control chart patterns: Type-2 fuzzy clustering optimized support vector machine″, ISA transactions. 2016. http://dx.doi.org/10.1016/j.isatra.2016.03.004

[12] M. Irfan, M. Usman, Yan Zhuang, Simon Fong, ″A critical Review of Security Threats in Cloud Computing″, Internation Symposium on Computational and Business Intelligence. Dec. 2015. https://doi.org/10.1109/iscbi.2015.26

[13] Mariano Graziano, Corrado Leita, Davide Balzarotti, ″Towards Network Containment in Malware Analysis Systems″, Annual Computer Security Applications Conference. 2012. https://doi.org/10.1145/2420950.2421000

◐ 저 자 소 개 ◑

배 원 일
2016년 목원대학교 컴퓨터공학과 졸업(학사)
2016년~현재 아주대학교 대학원 컴퓨터공학과 재학(석사)
관심분야 : 클라우드 컴퓨팅 보안, 암호프로토콜, 개인정보보호
E-mail : wibae.isaa@gmail.com

최 석 준
2016년 아주대학교 정보 및 컴퓨터공학과 졸업(학사)
2016년~현재 아주대학교 대학원 컴퓨터공학과 재학(석사)
관심분야 : 클라우드 보안, C-ITS 보안, 빅 데이터 보안.
E-mail : sjchoi.isaa@gmail.com

김 성 진
2010년 포항공과대학교 정보통신학 (석사)
2010년~현재 한국전자통신연구원 부설연구소 선임연구원
관심분야 : 클라우드 컴퓨팅 보안
E-mail : ksj1230@nsr.re.kr

김 형 천
2011년 고려대학교 정보보호대학원 (박사)
2001년~현재 한국전자통신연구원 부설연구소 책임연구원
관심분야 : 클라우드 컴퓨팅 보안
E-mail : khche@nsr.re.kr

곽 진
2000년 성균관대학교 졸업(학사)
2003년 성균관대학교 졸업(석사)
2006년 성균관대학교 졸업(박사)
2007~2015년 순천향대학교 정보보호학과 교수
2015년~현재 아주대학교 사이버보안학과 교수
관심분야 : 암호프로토콜, 개인정보보호, 정보보호제품평가, 클라우드 보안, 자동차 보안
E-mail : security@ajou.ac.kr