논문 2019-1-1 http://dx.doi.org/10.29056/jsav.2019.06.01

다수의 가상머신을 이용한 토르 트래픽 수집 시스템 설계 및 구현

최현재*, 김현수*, 신동명*†

Design and Implementation of Tor Traffic Collection System Using Multiple Virtual Machines

Hyun-Jae Choi*, Hyun-Soo Kim*, Dong-Myung Shin*†

요 약

본 논문에서는 사용자 및 서비스 제공자의 신원을 공개하지 않는 토르 네트워크상에서 불법적으로 콘텐츠를 공유하는 행위의 저작권 침해를 탐지하기 위하여 트래픽을 효율적으로 수집하고 분석하고자, 다수의 가상머신을 이용한 토르 트래픽 수집 시스템 설계 및 구현을 진행하였다. 토르 네트워크에 접속할 수 있는 클라이언트로 다수의 가상머신과 Mini PC를 이용하였으며, 스크립트 기반의 테스트 클라이언트 소프트웨어를 통해 트래픽 수집 서버에서 수집과 정제 과정을 모두 자동화하였다. 이 시스템을 통해 토르 네트워크 트래픽만을 저장하고 필요한 필드 데이터만을 데이터베이스에 저장할 수 있으며, 한 번의 수집 과정 당 평균적으로 약 10,000개 이상의 패킷을 데이터베이스에 저장하고 토르 트래픽만을 인식하여 정제하는 성능을 95% 이상 달성하였다.

Abstract

We intend to collect and analyze traffic efficiently in order to detect copyright infringement that illegally share contents on Tor network. We have designed and implemented a Tor traffic collection system using multiple virtual machines. We use a number of virtual machines and Mini PCs as clients to connect to Tor network, and automate both the collection and refinement processes in the traffic collection server through script-based test client software. Through this system, only the necessary field data on Tor network can be stored in the database, and only 95% or more of recognition of Tor traffic is achieved.

한글키워드 : 토르 네트워크, 가상머신, 트래픽 수집, 정제, 소프트웨어

keywords : tor network, virtual machine, traffic collection, refining, software

1. 서 론

1.1 토르 네트워크의 개념

토르(Tor; The Onion Routing)는 사용자의 출발지와 목적지의 IP 주소를 노출시키지 않고 인

* 엘에스웨어(주)
† 교신저자: 신동명(roland@lsware.com)
접수일자: 2019.05.30. 심사완료: 2019.06.15.
게재확정: 2019.06.20.

터넷을 이용할 수 있도록 설계된 기술로 현재 수천 개에 이르는 불특정 중계 노드를 통하여 패킷과 IP를 암호화하여 전달하는 방식으로 통신하고 있다. 토르 네트워크상에서는 일반적인 웹 사이트와는 다르게 히든 서비스(Hidden Service)라는 서비스를 사용자들에게 제공을 하는데 .onion이라는 주소를 사용하며, 토르 브라우저만으로 접속할 수 있다. 토르 브라우저를 이용하여 웹 사이트에 접속할 때는 최소 3종류의 중계 노드(Entry Node, Relay Node, Exit Node)를 경유하며, 히든 서비스에 접속할 때는 최소 6개의 중계 노드를 경유하여 해당 목적지에 접속하게 된다[1]. 히든 서비스에 접속할 때 사용자는 첫 번째 경유하는 3개의 중계노드들의 IP주소는 알 수 있지만 두 번째 경유하는 3개의 중계노드들의 IP주소는 알 수가 없다. 동일하게 히든 서비스 입장에서도 서버와 연결되는 첫 번째로 경유하는 3개의 중계노드들의 IP주소는 알 수 있지만, 히든 서비스에 접속하려는 사용자와 연결된 3개의 중계노드들의 IP주소는 알 수 없다. 그렇기 때문에 디렉터리 서버는 두 번째 경유 하게 되는 3개의 중계노드들과 연결시켜주는 역할을 하며, 3개의 노드와 3개의 노드가 만나는 지점을 랑데부 노드(Rendezvous Node)라고 한다. 노드들은 전달 받은 데이터를 각자의 키로 암호화하고 접속자와 서버 모두 어떤 노드를 경유했는지 알 수 없기 때문에 토르 네트워크는 익명성이 보장되며 추적이 어렵다고 할 수 있다[2].

1.2 문제점 및 연구 내용

토르 네트워크는 사용자나 서비스 제공자의 신원이 숨겨져 알 수 없기 때문에 마약이나 무기 등의 불법적인 거래에 악용되어 그 이용률이 급속히 증가하고 있으며, 콘텐츠를 불법 유통시키는 새로운 채널로 문제가 되고 있는 상황이다. 앞으로 토르 네트워크상에서 불법적으로 콘텐츠를 공유하는 행위인 저작권 침해 사례가 심각해짐에 따라 토르 네트워크상에서도 저작권 침해를 탐지하기 위한 기술을 필요로 한다.

본 연구에서는 저작권 침해를 탐지하기 위한 선행 연구로 대량의 토르 네트워크 트래픽을 효율적으로 수집하고자 한다. 첫 번째로 토르 트래픽 수집을 위한 시스템 구성과 수집을 위해 사용한 모듈, 트래픽을 저장하는 데이터베이스 설계를 소개할 것이며, 두 번째로 수집 시스템 자동화를 위해 개발한 유틸리티 트래픽 필터링 방법을 설명하고 마지막으로 성능 평가와 결론으로 마무리하고자 한다.

2. 토르 트래픽 수집 과정

2.1 토르 트래픽 수집 시스템

토르 트래픽을 수집하기 위해 준비한 장비의 구성도는 그림 1과 같다.

토르 네트워크 트래픽 수집을 위한 방화벽, IDS, IPS 등에 의해 트래픽이 차단되지 않도록 별도의 토르 네트워크 트래픽 수집용 네트워크 망을 구축하였으며, 가상머신 PC는 Xen Server v7.5 OS를 설치하였다. 가상머신 PC 내부에는 독립적으로 동작하는 20대의 가상머신을 Ubuntu v16.04 TLS로 설치하여 토르 네트워크에 접속하는 클라이언트로 이용하였다. 또한 가상머신을 이용하여 생길 수 있는 문제를 보완하기 위하여 Mini PC 5대에 동일한 OS를 설치하여 클라이언트로 이용하였다.

각각의 가상머신과 Mini PC는 우리가 구축한 토르 네트워크 내에서 사설 IP를 할당받아 고정하고 토르 네트워크에 접속한다. 이때 접속하는 토르 네트워크의 첫 번째 Entry Node는 대역폭

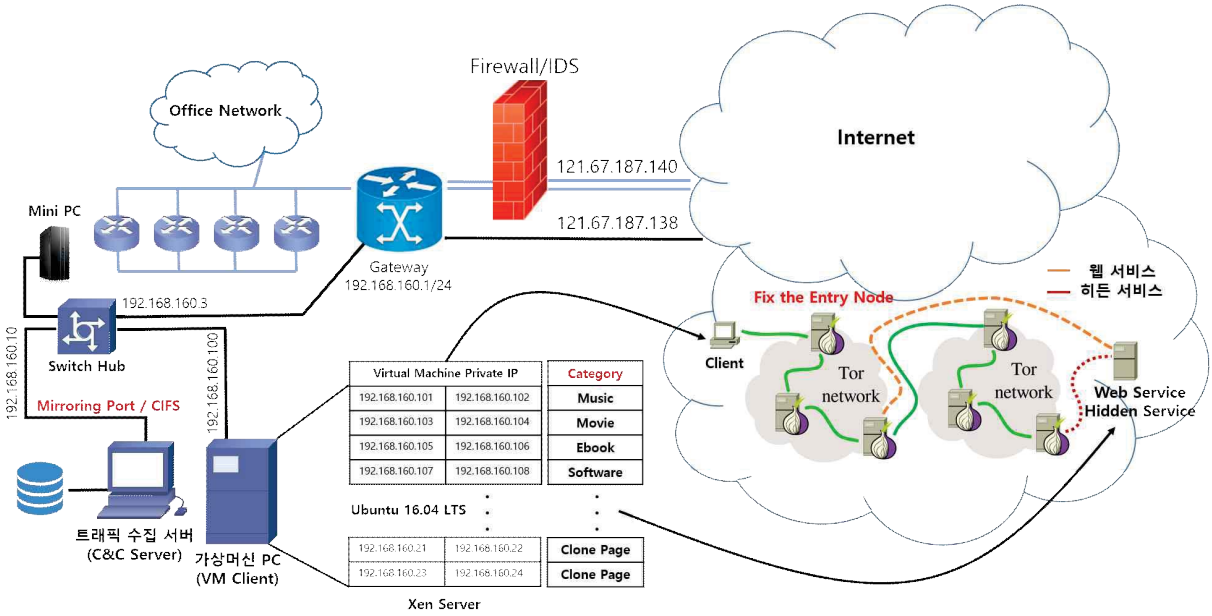

그림 1. 토르 네트워크 트래픽 수집 시스템 구성도
Fig. 1. Tor network traffic collection system configuration

과 속도를 고려하여 히든서비스의 성격에 맞게 음악, 영화, 이북, 소프트웨어 등과 같게 카테고리 별로 분류하였다.

트래픽 수집 서버는 가상머신 PC내의 가상머신과 Mini PC들을 일괄적으로 통제하도록 하였다. 토르 네트워크에 접속하는 가상머신들에게 노드 설정 및 트래픽 수집 등의 명령을 내려 트래픽을 수집한다. 트래픽 수집 방법은 가상머신들이 각각 수집하여 트래픽 수집 서버로 전달하는 방법과 트래픽 수집 서버를 스위치 미러링 포트에 연결하여 가상머신들의 전체 트래픽을 수집하는 방법을 사용하였다.

2.2 트래픽 수집 데이터 저장을 위한 데이터베이스 설계

트래픽 수집 시스템을 이용하여 수집한 토르 트래픽 데이터는 추후 특징점 연구에 이용할 수

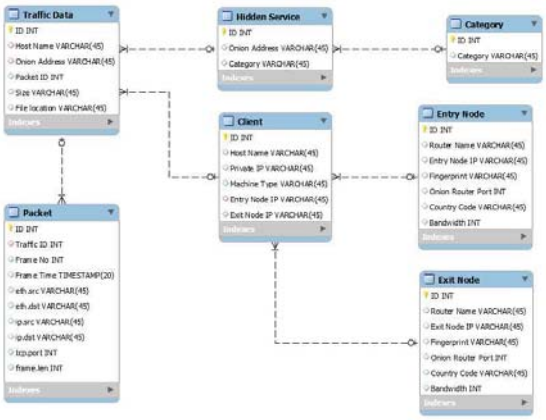

그림 2. 토르 네트워크 트래픽 데이터 테이블 정의
Fig. 2. Tor network traffic data table definition

있도록 필요한 필드만 정제하여 저장한다. 그림 2는 정제된 트래픽 데이터를 저장할 수 있는 데이터 테이블의 구조이다. 가상머신들을 통해 트래픽 수집 서버에 저장된 pcap 파일들은 필요한 필드만 csv 데이터로 변환 후 PostgreSQL 데이

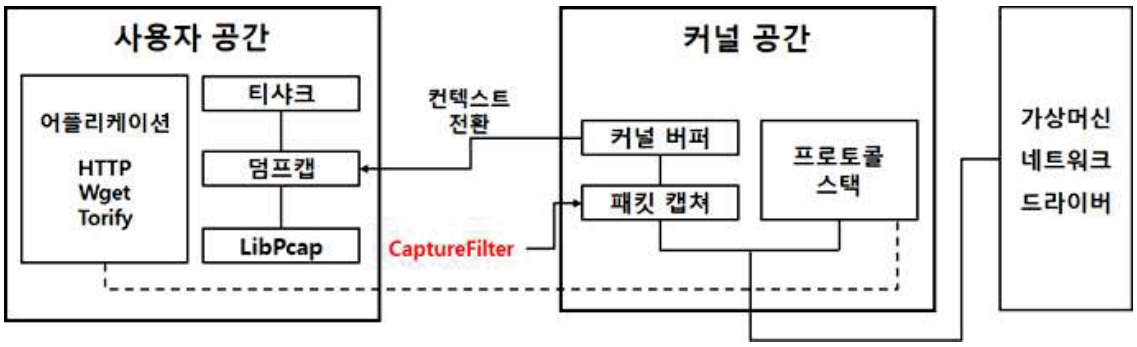

그림 3. Tshark를 이용한 트래픽 수집 및 필터링
Fig. 3. Traffic collection and filtering using Tshark

터베이스에 저장하였다.

Traffic Data 테이블은 특정 가상머신 또는 특정 히든 서비스를 기준으로 트래픽 데이터의 요약된 정보를 확인할 수 있으며 Packet 테이블은 특정 히든 서비스를 한번 접속했을 때 수집할 수 있는 트래픽의 시간, MAC, IP, Port, 패킷 길이 등을 기준으로 상세하게 저장하고 조회할 수 있다.

Hidden Service 테이블은 토르 네트워크상에서 불법 콘텐츠를 유통하는 히든 서비스의 주소를 저장하며 Category 테이블을 통해 불법 콘텐츠를 유통하는 히든 서비스를 유형별로 조회할 수 있다.

마지막으로 Client 테이블은 각각의 가상머신들의 정보를 저장하며 Entry Node 테이블과 Exit Node 테이블을 이용하여 특정 가상머신이 어떤 Node를 사용했는지 해당 Node의 정보를 조회할 수 있다.

2.3 트래픽 수집을 위해 사용하는 소프트웨어 모듈

가상머신 상에 토르 패키지(torify, torsocks), wget, Tshark 모듈을 설치하여 트래픽 수집에 사용하였다.

torify 모듈은 시스템에서 사용할 수 있는 가장 기본적이고 단순한 Tor 래퍼로 특정 구성파일(/etc/tor/torrc 파일 내에서 수정하여 Entry Node를 변경)을 사용하여 torsocks를 호출한다.

torsock 모듈은 LD_PRELOAD라는 환경변수를 이용하여 libtorsocks.so 라이브러리를 bash 상에 업로드하고 해당 bash는 Tor를 이용한 암호화 통신을 수행한다. 본 논문의 트래픽 수집 과정에서는 클라이언트의 리소스를 많이 소모하는 토르 브라우저를 사용하지 않고 torify를 통해 torsock을 호출하여 토르 네트워크에 접속하는 방식을 사용하였다.

wget 모듈은 웹 서버로부터 콘텐츠를 가져오는 소프트웨어로 HTTP, HTTPS, FTP 프로토콜 등을 이용하여 다운로드를 지원한다.

Tshark 모듈은 네트워크 드라이버에 네트워크 패킷이 수신될 때마다 패킷의 사본을 필터링하여 커널 서브시스템에 전송하는 소프트웨어로 libpcap 파일 형식으로 저장을 한다.

2.4 트래픽 수집 및 정제를 위한 유틸리티 개발

토르 네트워크 트래픽을 수집하기 위해 Tshark 모듈을 가상머신과 Mini PC 그리고 트래픽 수집 서버에 모두 설치한다. 또한 안정적인

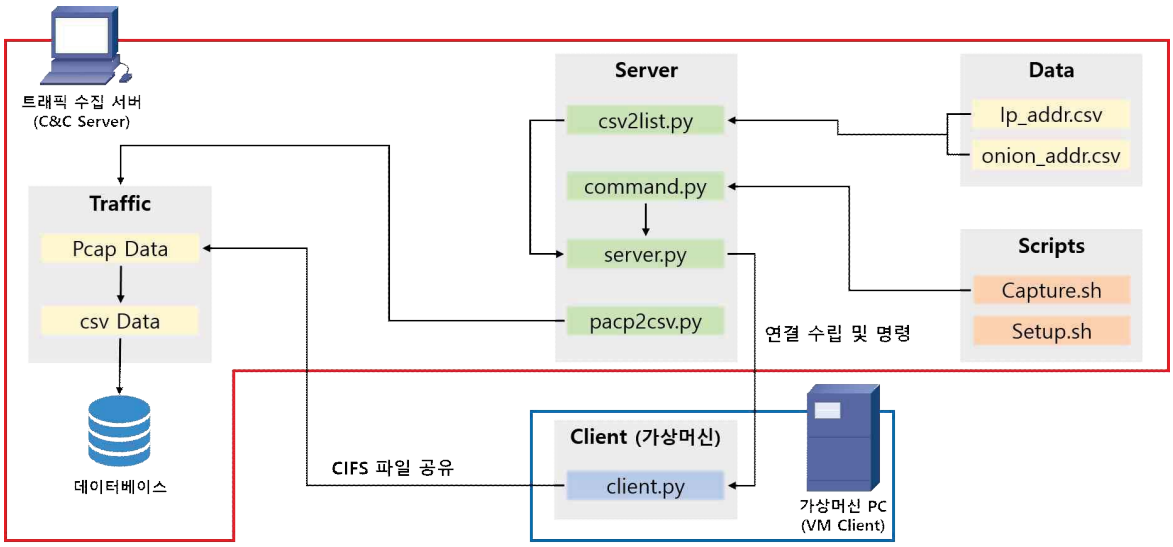

그림 4. Tshark를 이용한 트래픽 수집 및 필터링
Fig. 4. Script-based test client software configuration diagram

속도와 지연율로 트래픽을 수집하기 위해 Tor Network Status (https://torstatus.blutmagie.de)를 참고하여 높은 대역폭을 가진 노드를 Entry Node로 고정시킨다[3][4]. (/etc/tor/torrc 파일 내에서 수정) 모든 클라이언트에서 노드 설정이 완료되면, Tshark 모듈로 패킷 수집을 시작한다. Tshark 모듈은 패킷 수집과 정제 과정을 동시에 수행하기 위하여 그림 3과 같이 CaptureFilter에 Entry Node의 IP와 ORPort를 필터링 옵션으로 설정하여 트래픽을 수집한다[5].

CaptureFilter는 수집 당시 설정한 필터링 옵션에 따라 패킷을 필터링하여 덤프캡으로 전송하며, 이 덤프캡은 전송된 패킷을 가상머신 저장공간에 libpcap 파일 형식으로 저장을 한다. 만약 필터링 옵션이 없이 가상머신의 모든 네트워크 트래픽을 모두 수집하여 정제하는 것은 수집할 때 모든 패킷을 커널 공간에서 사용자 공간의 덤프캡으로 복사(컨텍스트 전환) 해야 함으로 CPU에 많은 부하를 주게 된다[6]. 따라서 수집 과정과 필터링 과정을 크게 분리하지 않고 CaptureFilter 옵션을 사용하여 커널 공간에서 토르 네트워크 트래픽만 사용자 공간으로 복사를 하도록 한다. 본 논문에서는 수집과 정제 과정을 하나의 쉘 스크립트로 작성하여 가상머신과 Mini PC에서 동작할 수 있도록 개발하였다[7][8].

3. 수집 시스템 자동화 및 성능 평가

본 논문에서는 토르 네트워크에 접속하는 다수의 가상머신과 Mini PC들을 일괄적으로 관리하고 명령을 내릴 수 있는 Command & Control 역할을 하는 트래픽 수집 서버를 개발하였다. 동작 과정과 구성은 그림 4와 같다.

20대 이상의 가상머신들과 Mini PC들이 트래픽 수집 서버와 연결을 수립하고 접근 허용과 명령을 이해할 수 있는 클라이언트 프로그램을 통해 첫 번째로 Entry Node 및 Exit Node를 변경할 수 있는 torrc 파일을 변경할 수 있으며, 두 번째로 torify 모듈과 wget 모듈을 이용하여 토

르 네트워크 접속 및 웹 페이지 다운로드를 할 수 있으며, 세 번째로 토르 네트워크 접속 및 웹 페이지 다운로드 과정과 동시에 Tshark 모듈을 이용하여 토르 네트워크 트래픽 수집 및 정제를 할 수 있다. 마지막으로 트래픽 수집 서버는 토르 네트워크 트래픽이 저장된 pcap 형식의 파일을 csv 형식의 시계열 데이터로 변환하고 변환된 csv 형식의 데이터를 PostgreSQL 데이터베이스로 저장할 수 있는 프로그램을 개발하였다. 각 프로그램의 역할은 아래와 같다.

- setup.sh

Entry Node 및 Exit Node를 변경할 수 있는 스크립트로 torrc 파일 제어한다. (Node를 식별할 수 있는 Fingerprint 값 명시)

- ip_addr.csv

동작을 수행할 가상머신들의 내부 IP 주소를 입력한다.

- onion_addr.csv

트래픽을 수집 대상이 될 토르 네트워크상의 히든 서비스(Onion 주소) 입력한다.

- client.py

각각의 가상머신과 Mini PC에서 client.py를 실행하여 트래픽 수집 서버로부터 명령받은 동작을 수행할 수 있도록 대기한다. (Listening 상태)

- command.py

트래픽 수집 서버에서 server.py를 입력했을 때, 가상머신들은 해당 아래 내용들을 수행하도록 한다.

capture.sh는 torify 모듈을 통해 토르 네트워크에 접속하고 wget을 통해 히든 서비스 페이지를 다운로드 이와 동시에 Tshark 모듈을 이용하여 토르 네트워크 트래픽 수집 및 정제 스크립트를 실행시키고 가상머신에서 트래픽 수집이 완료되면 CIFS 공유를 통해 트래픽 수집 서버에 트래픽을 저장한다.

- server.py

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
23.53.225.247	175	191 k	121	187 k	54	4309
125.209.222.142	10	1258	4	639	6	619
178.132.78.148	371	399 k	211	367 k	160	32 k
192.168.160.10	5,389	4123 k	3,885	1209 k	1,504	2914 k
192.168.160.200	6,468	5205 k	1,946	2966 k	4,522	2238 k
203.133.167.16	250	239 k	145	233 k	105	5968
211.231.99.17	10	1254	4	636	6	618
216.58.199.100	229	244 k	136	237 k	93	7419
216.58.199.110	10	1492	4	812	6	680

그림 5. 정제된 토르 네트워크 트래픽 데이터
Fig. 5. Refined tor network traffic data

트래픽 수집 서버에서 server.py를 실행하여 client.py를 실행한 가상머신들에게 동작을 명령한다. onion_addr.csv에 입력된 히든 서비스를 대상으로 ip.addr.csv에 입력된 가상머신들에게 command.py에 작성된 명령을 전달하여 트래픽 수집을 하도록 한다.

- pcap2csv.py

트래픽 수집 서버에 저장된 트래픽 파일(.pcap)을 csv 형식 데이터로 변환한다.

위 스크립트 기반의 테스트 클라이언트 소프트웨어를 통해 수집 과정을 자동화할 수 있었으며, 한 번의 수집 과정 당 평균적으로 약 10,000개 이상의 패킷을 데이터베이스에 저장하고 그림 5와 같이 토르 트래픽만을 인식하여 정제하는 성능을 95% 이상 달성하였다.

4. 결 론

토르 네트워크는 사용자나 서비스 제공자의 신원이 숨겨져 알 수 없기 때문에 마약이나 무기 등의 불법적인 거래뿐만 아니라 콘텐츠를 불법 유통의 경로로 사용되고 있어 불법 복제물 유통으로 인한 저작권 침해 문제가 증가하고 있는 상황이다.

본 논문에서는 다수의 가상머신을 이용한 토르 트래픽 수집 시스템을 통해 수집과정을 자동화하고 높은 정제율로 토르 네트워크 트래픽만을

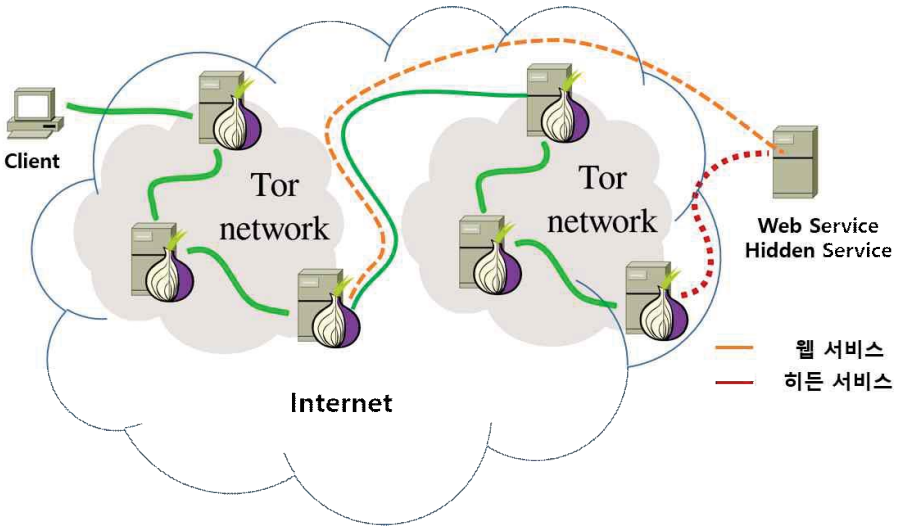

그림 6. 웹 서비스와 히든 서비스를 동시에 수행하는 서버의 네트워크 구성도
Fig. 6. Network configuration of server that plays role as a web service and a hidden service at the same time

저장할 수 있었으며, 필요한 필드 데이터만을 데이터베이스에 저장할 수 있음을 보여주었다.

향후 계획으로는 첫 번째로 그림 6과 같이 테스트를 위한 히든 서비스 수행 서버를 개발할 예정이다. 웹 서비스와 히든 서비스 두 가지 서비스를 수행할 수 있는 서버를 Google Cloud Platform VM 인스턴스에 구축하여 트래픽의 차이를 비교 분석하고 웹 서비스의 트래픽을 통해 히든 서비스의 컨텐츠를 유추할 수 있는 방법을 연구할 예정이다. Google Cloud Platform VM 인스턴스에 서버를 구축하는 이유는 트래픽 수집 시스템의 가상머신들과 다른 공인 IP를 사용할 수 있기 때문이다. 플랫폼 자체에서 지역에 따른 공인 IP를 유동적으로 변경할 수 있기 때문에 IP가 변경될 때마다 바뀌게 되는 중계노드에 따른 트래픽의 차이 또한 수집할 수 있을 거라고 예상된다[9].

두 번째로 트래픽 수집 시 진입 노드 고정과 사용 모듈 그리고 수집 소요 시간에 따른 트래픽 차이에 대한 고찰이다. 앞서 설명한 연구에서는 진입 노드인 Entry Node를 선별하여 고정하고 wget 모듈을 사용하며 접속한 페이지의 콘텐츠가 모두 로드될 때 질 때까지 수집하였다. 하지만 실제 토르 브라우저를 사용하는 사용자들은 진입 노드를 고정하지 않고 wget 모듈도 사용하지 않기 때문에 현재 연구에서 수집하는 트래픽과 어떤 차이가 있는지 연구할 예정이다.

마지막으로 수집한 트래픽을 이용하여 추후 트래픽 패턴 분석과 기계학습을 이용한 특징점 분석 그리고 트래픽 액티브, 패시브 트래픽 핑거프린팅(Traffic Fingerprinting)기술 등의 연구를 진행할 예정이다[10].

Acknowledgement

본 연구는 문화체육관광부 및 한국저작권위원회의 2019년도 저작권기술개발사업의 연구결과로 수행되었음

참 고 문 헌

[1] Dingledine, Roger, Mathewson, Nick, and Syverson, Paul, “Tor: The Second-generation Onion Router”, Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13, pp.1-17, 2004. https://doi.org/10.21236/ada465464

[2] Nathan Evans, Roger Dingledine, and Christian Grothoff, “A practical congestion attack on Tor using long paths”, USENIX Security Symposium, 2009. https://dl.acm.org/citation.cfm?id=1855771

[3] Frank Cangialosi, Dave Levin, and Neil Spring, “Ting: Measuring and Exploiting Latencies Between All Tor Nodes”, Proceedings of the 2015 Internet Measurement Conference, Oct. 2015. https://dl.acm.org/citation.cfm?id=2815701

[4] Wang, Tao and Ian Goldberg, “Improved website fingerprinting on tor”, Proceedings of the 12th ACM workshop on privacy in the electronic society, ACM, 2013. https://dl.acm.org/citation.cfm?id=2517851

[5] Yossi Gilad and Amir Herzberg, “Spying in the dark: TCP and Tor traffic analysis”, Proceedings of the 12th Privacy Enhancing Technologies Symposium (PETS 2012), Jul. 2012. https://doi.org/10.1007/978-3-642-31680-7_6

[6] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, and Jennifer Rexford, “RAPTOR: Routing Attacks on Privacy in Tor”, Proceedings of the 24th USENIX Security Symposium, Aug. 2015. https://arxiv.org/abs/1503.03940

[7] Lashkari, Arash Habibi, et al., “Characterization of Tor Traffic using Time based Features”, ICISSP, 2017. https://doi.org/10.5220/0006105602530262

[8] Rimmer, Vera, et al. “Automated website fingerprinting through deep learning”, arXiv preprint arXiv:1708.06376, 2017. https://doi.org/10.14722/ndss.2018.23105

[9] Montieri, Antonio, et al., “Anonymity services Tor, I2P, JonDonym: Classifying in the dark”, 9th International Teletraffic Congress (ITC 29), Vol. 1, IEEE, 2017. https://doi.org/10.23919/itc.2017.8064342

[10] Hayes, Jamie and George Danezis, “k-fingerprinting: A robust scalable website fingerprinting technique”, 25th {USENIX} Security Symposium ({USENIX} Security 16), 2016. https://dl.acm.org/citation.cfm?id=3241186

저 자 소 개

최현재(Hyun-Jae Choi)

2018년 성균관대학교
전자전기컴퓨터학과 석사
2018년-현재 엘에스웨어(주) 주임

<주관심분야> 네트워크/시스템 보안, 취약점분석, 블록체인

김현수(Hyun-Soo Kim)

2019년 단국대학교
소프트웨어학과 학사
2019년-현재 엘에스웨어(주) 주임

<주관심분야> 소프트웨어 공학, 소프트웨어 테스팅, 네트워크, 인공지능

신동명(Dong-Myung Shin)

2003년 대전대학교 컴퓨터공학과 박사
2001년-2006년 한국정보보호진흥원
응용기술팀 선임연구원
2006년-2014년 한국저작권위원회
저작권기술팀 팀장
2014년-2016년 한국스마트그리드사업단
보안인증팀 팀장
2016년- 현재 엘에스웨어(주)
연구소장/상무이사

<주관심분야> 오픈소스 라이선스, 시스템/네트워크보안, SG인증/보안, SW취약점분석・감정, 블록체인