Journal of Korea Multimedia Society Vol. 20, No. 6, June 2017(pp. 911-917)
https://doi.org/10.9717/kmms.2017.20.6.911

출구 릴레이 방법론을 이용한 Tor 보안 정책

장덕성[†], 박소연[††], 최두영[†††]

A Tor Security Policy using Exit Relay Methodology

Duk-Sung Jang[†], So-Yeon Park[††], Du-Young Choi[†††]

ABSTRACT

Tor proxy tool is studied which is most frequently used for ransomeware to penetrate into sensitive information. It will be researched for the malicious methods to spread virus by using Tor and considered a countermeasure to prevent them. We present exit relay methodology for Tor security policy, simulate it, and get a probability to detect the ransomeware. And we compare it with TSS technology which is able to protect the attack via virtual server on exit relay.

Key words: Tor Network, Ransomeware, Mirror Server, Exit Relay

1. 서 론

랜섬웨어(ransomeware)는 상대방 PC의 파일을 암호화 한 뒤 비트 코인을 요구하는 악명 높은 바이러스이다[1]. 그것은 대부분 Tor를 통하여 유포되며 해당 사이트에 접속하거나 광고를 클릭하면, drive-by-download 형식으로 피해 PC에 심어지게 된다. 랜섬웨어 뿐만 아니라 다른 악성 프로그램들도 검거를 피하기 위해 익명성이 보장된 Tor를 통하여 바이러스들을 유포시키고 있다. 이러한 이유들 때문에 Tor는 바이러스와 악성 프로그램들을 걸러내는 거름망 장치가 꼭 필요하다[2]. Tor는 온라인상에서 트래픽 분석이나 IP주소 추적을 불가능하게 하여 익명성을 보장하는 네트워크(network)이며, 인터넷에서 개인 프라이버시를 보호하려는 목적으로 많은 사용자들에 의해 사용된다. 트래픽은 목적지까지 바로 전달되지 않고, 네트워크 속 임의의 중계 서버(server)를 통해 전송되기 때문에 패킷(packet)의 출발지와 목적지가 어딘지 알 수 없다. 전송되는 패킷은 여러 겹의 암호화가 되어서 전송되며 중간 서버를 거칠 때마다 한 겹씩 복호화 되어 다음 서버에 전달된다[3].

Tor 네트워크는 크게 입구 릴레이(guard relay), 중간 릴레이(middle relay), 출구 릴레이(exit relay)로 구성되어 있다[4]. 그리고 Tor 네트워크 이용 시 패킷은 겹겹이 암호화가 되고, 이 패킷들은 한 겹씩 라우터를 거치면서 복호화가 된다. 노드를 거칠 때마다 패킷을 둘러싸고 있는 암호화들이 알맞은 공개키에 의하여 양파가 벗겨지듯이 복호화가 일어나게 된다. 그래서 이것을 양파 라우터(onion router) 라고 부른다[5]. Tor의 장점에도 불구하고 익명성 때문에 Tor를 이용한 악용 사례가 늘고 있다. 대표적인 것으로 랜섬웨어, DDOS 공격, Macro 악성코드 유포 등이 있다[6]. Tor의 보안은 '익명성 보장' 이란 장점을

※ Corresponding Author : Duk-Sung Jang, Address: (704-701) 1000 Shindang-Dong, Dalseo-Gu, Daegu, Korea , TEL : +82-53-580-5165, FAX : +82-53-580-5267, E-mail : dsjang@kmu.ac.kr
Receipt date : Mar. 30, 2017, Revision date : May 10, 2017
Approval date : May 25, 2017
[†] Dept. of Computer Engineering, Keimyung University
[††] Dept. of Computer Engineering, Keimyung University (E-mail : soyen5105@naver.com)
[†††] Corea Information Technology Inc. (E-mail : dychoi@cit21.com)
※ This work was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2016(Grants No. C0397109).

살리면서, 사용자들이 원활하고 안전하게 Tor를 이용할 수 있도록 하여야 한다[7].

본 논문에서는 Tor 보안 정책의 일환으로 출구 릴레이에 미러 서버(mirror server) 를 설치하여 패킷을 확인하고, 악성 코드를 걸러내는 방법을 다루게 된다. 수천 개의 패킷을 처리할 수 있도록 시뮬레이터(simulator)를 만들고 악성 바이러스 수천 개를 숨겨 그 중 몇 %를 검출할 수 있는 지를 실험하였다. 미러 서버를 두는 위치에 따라 입구 릴레이, 중간 릴레이, 출구 릴레이가 있다고 했는데, 출구 릴레이에 미러 서버를 설치하는 방법을 줄여서 출구 릴레이 방법론(Exit Relay Methodology)이라 하기로 한다. 우리가 제안하는 출구 릴레이 방법론은 기존의 TSS (Tor Support Servers) 기술[8]과 유사한 면이 있다. TSS는 출구 릴레이에 가상의 서버를 설치하여 Tor의 보안을 향상시키고자 하는 기술이다.

2장에서는 TSS에 대해 설명하고, 3장에서는 출구 릴레이 방법론에 대해 자세히 기술한다. 4장에서 출구 릴레이 방법론을 시뮬레이션해서 평가하고, 5장에서 결론을 맺는다.

2. Tor Support Servers

Tor의 시스템은 패킷을 보낼 때, 사용자의 패킷이 손실되지 않는 포트를 찾아 최적의 경로를 선택한다. 전송해 줄 포트의 라우터 수를 기준으로 선택 경로를 판단하게 되는데, 사용되는 라우터가 상대적으로 적은 포트가 우선 선택된다. Fig. 1은 각 라우터의 출구 정책에 따른 라우터들을 그룹화한 것이다. 80번 포트를 지나는 패킷을 전달해주는 라우터가 라우터 1, 2, 3, 4이고, 21번 포트를 지나는 패킷을 전달하는 라우터는 라우터 1, 5이다. 만약 공격자가 21번 포트를 지나는 패킷을 공격 목표로 삼는 경우 다른 라우터는 신경 쓸 필요 없이 1, 5 라우터만 공격하면 된다. 이와 같은 공격을 분할 공격이라고 한다[8].

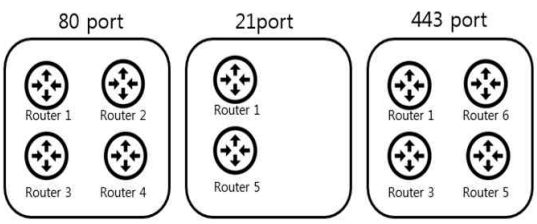

Fig. 1. Grouping of Routers by Allocating to Ports.

Tor의 출구 릴레이 관리자가 TSS를 운영하여 필요에 따라 출구 정책을 바꾼다면 분할 공격에 대응할 수 있다. 예를 들어 21번 포트를 지나는 패킷을 전달하는 라우터의 수가 다른 포트를 서비스하는 라우터의 수보다 적다면 Fig. 2와 같이 TSS가 가상서버 두 개를 더 할당하여 라우터 9와 10을 21번 포트에 배정하는 것이다. 이 경우 라우터 9와 10를 TSS 서버라고 한다. 이렇게 각 포트를 지나는 패킷을 서비스하는 라우터의 수에 차이가 없도록 하는 것이 TSS의 분할 공격 대응 방법이다.

공격의 대상이 되는 포트의 라우터 수를 A개라 가정하고, 그 외 공격대상에서 제외된 포트들 중 라우터의 수가 가장 많은 포트의 라우터 수를 B개라고 가정하면, A개의 라우터를 가진 포트가 공격을 회피할 수 있는 회피확률은 (B-A) / A 이다.

회피확률은 공격확률과는 반대되는 개념이다. TSS 바이러스 공격확률을 구해보자. 예를 들어 Fig. 3과 같이 80번 포트는 8개의 라우터와 연결이 되어있고, 21번 포트는 2개의 라우터와 연결이 되어있다고 가정 했을 때, 공격자는 상대적으로 적은 라우터와 연결되어 있는 21번 포트에 이용자들이 몰릴 것이라 예상하게 된다. 그러므로 21번 포트가 80번이나 443번 포트 보다 라우터들이 공격당할 확률이 그만큼 높아진다. 각 포트가 공격받을 확률은 80번 포트는 1/8, 21번 포트는 1/2, 443번 포트는 1/4 으로써, 21번 포트가 가장 높다.

만약 Fig. 4와 같이 21번 포트에 가상의 라우터 서버를 6개 더 할당하게 되면, 공격당할 확률이 1/8로 낮아지게 되고, 공격 회피률은 6/8이 된다. 각 포트별 공격을 당할 확률은 각각 1/8, 1/8, 1/4로서 어떠한 포트도 공격을 당할 확률이 50%가 넘지 않게 되어 전체적인 보안의 향상을 기대할 수 있다. 이와 같이 출구 릴레이에 TSS 서버를 운영하여 Tor의 보안을 향상시키고자 하는 것이 TSS 정책이다. 그러나

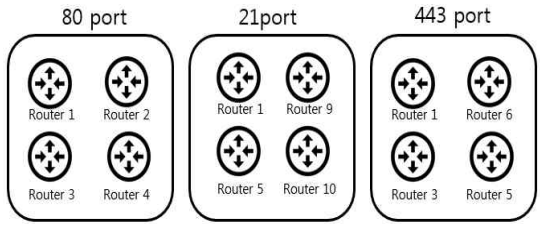

Fig. 2. Adding some Routers by TSS Exit Policy.

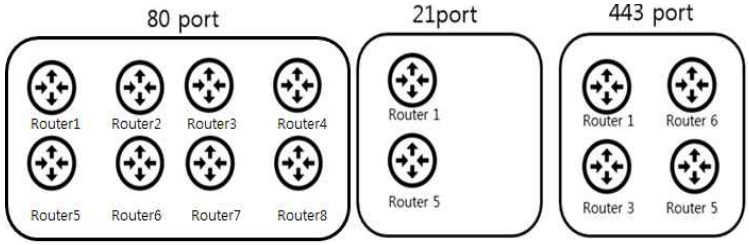

Fig. 3. Before Allocating Virtual Router to 21 Port.

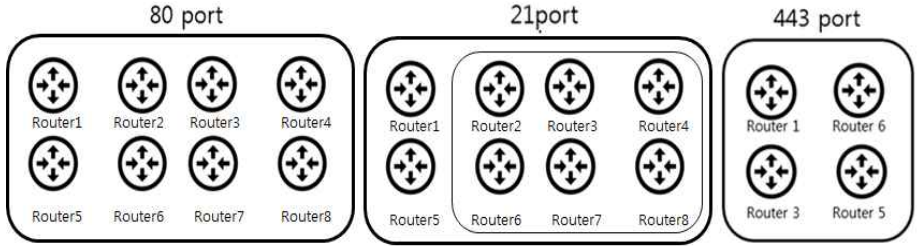

Fig. 4. After Allocating Virtual Router to 21 Port..

공격자가 TSS 서버의 존재를 알고 공격 대상에서 TSS 서버를 제외시켜버리면, TSS 정책은 실패하게 된다.

3. 출구릴레이 방법론

출구 릴레이에 가상서버를 운영하여 보안을 향상시키는 방법이 TSS 정책이다. 그러나 TSS는 바이러스를 제거하기 위한 방법은 아니다. 우리는 TSS의 보안 정책과는 다른 새로운 방법으로, 출구 릴레이를 찾아서 거기에 미러 서버를 두고 바이러스를 검출하는 새로운 출구 릴레이 방법론(Exit Relay Methodology)을 제안하고자 한다. 출구 릴레이에 대한 공격 사례는 레드삭스 해킹 팀의 역추적 사례[9]와 아키마이 양파 라우터 프로젝트[10]에서 그 예를 찾아볼 수 있다.

출구 릴레이 방법론을 설명하기 전에 먼저 가드 릴레이 방법론부터 설명하고자 한다. 가드 릴레이 방법론은 패킷이 첫 번째 라우터와 연결되어 있는 가드 릴레이에 미러 서버를 설치하는 방법이다. 모든 패킷들이 원하는 목적지에 도달하기 위해서는 반드시 가드 릴레이를 지나야하기 때문에 Tor를 통하여 전송되는 모든 패킷을 검열할 수 있다. 그러므로 가드 릴레이 방법론을 사용하면 바이러스를 검출확률은 100%이다. 그러나 미러 서버는 패킷의 복호화 키를 가져야 하고, 패킷의 복호화 키는 디렉토리 서버에서 가지고 있으며, 디렉토리 서버는 패킷이 특정 라우터에 도달 하였을 때, 라우터의 키와 패킷의 복호화 키를 일대 일로 교환하는 방식을 취하기 때문에, 현실적으로 가드 릴레이의 미러 서버가 미리 복호화 키를 갖는 것은 불가능하다. 가드 릴레이 방법론을 구현하기 불가능하므로 대안으로 출구 릴레이 방법론을 제안하고자 한다.

출구 릴레이 방법론이 성립하기 위해서는 출구 릴레이를 찾기 위한 경로 탐색 방법이 필요하다. 우리는 캐스케이딩(cascading) 방식[11]을 경로 탐색 방법으로 채택하였다. 캐스케이딩 방식은 추천된 여러 경로 중 하나의 경로를 선택하여 메시지를 전달하는 방식이다. 캐스케이딩 방식에서의 경로 추천은 통계적 방법을 따른다. Tor를 이용하여 전달된 과거의 패킷들의 경로 정보를 수집하여, 라우터의 대역폭이나 패킷의 밀집률을 분석한다. 1Mbps당 1M 바이트 크기의 패킷을 처리하는데 1초가 걸리는 것으로 가정했을 때, 만약에 대역폭이 100Mbps이고 패킷이 200개가 들어왔다면, 2초가 걸릴 것으로 추정할 수 있다. 그러나 다음 릴레이의 대역폭이 30Mbps 라면 200개의 패킷을 처리하는 데는 3.3배의 시간이 더 소모되어 6.6초가 걸릴 것으로 추정된다. 그만큼 지연(delay)이 일어나는 것이므로 캐스케이딩 방식에서

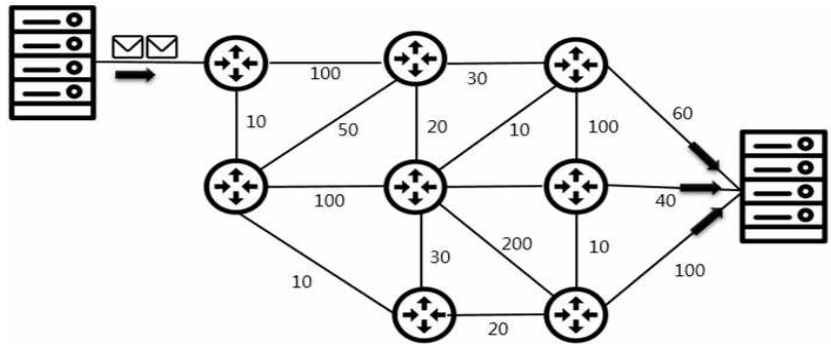

Fig. 5. Statistical Analysis of Cascading Network Routing.

는 이런 경로는 추천하지 아니한다. 그러나 공식에 의한 것이 아니고 통계적으로 가장 많이 사용된 릴레이 중 최선의 릴레이와 차선의 릴레이 등 우선순위를 부여하여 경로를 추천하고, 최선의 경로에 부하가 많으면 차선의 경로를 선택하게 하는 방식이다.

Fig. 5는 추천된 라우팅 맵이며 노드는 라우터이고 간선의 숫자는 대역폭이다. 그러나 이 라우팅 맵이 최종 경로를 결정하지는 못한다. 만약 암호화 겹수가 3겹이면 추천된 라우팅 맵 중 어느 라우터인지 알 수 없지만 복호화를 위해서 반드시 3개의 라우터를 거쳐서 최종 출구 릴레이로 전달되도록 방향을 잡기 때문이다. 확실한 것은 캐스케이딩 방식에 의거하여 출구 릴레이를 구할 수 있으므로 여기에 미러 서버를 두면 된다. Fig. 5에서는 60, 40, 100이라고 표기된 간선이 출구 릴레이가 된다.

4. 실험 및 평가

출구 릴레이 방법론을 시뮬레이션하기 위한 알고리즘은 Algorithm 1과 같다. Routing(flag)은 패킷이 도착하면 그것을 출구 릴레이까지 전달하는 함수이다. 패킷이 유입되었다는 것은 flag로 나타낸다. 패킷이 도착하지 않으면 계속해서 패킷을 기다리게 되고, 패킷이 도착하면 Cascading(packet)을 호출한다. Cascading()은 3장에서 설명한 바와 같이 Tor 네트워크의 대역폭과 과거의 패킷 밀집률로부터 통계적으로 라우팅 맵을 구하고, exit_relays라는 리스트를 반환한다. exit_relays에는 출구 릴레이들이 차례로 들어있고, 이들 출구 릴레이 각각에 미러 서버를 두고 Mirror(exit)을 통해서 바이러스가 있으면 바이러스를 제거하게 된다.

출구 릴레이의 방법론에서는 미러 서버가 디렉토리 서버에서 복호화 키를 받아오지 않아도 된다. 이미 미들 릴레이를 거치는 동안에 복호화가 완료되고 출구 릴레이에서는 복호화가 완료된 데이터만 남게 되기 때문이다.

Algorithm 1을 기초로 출구 릴레이방법론을 시뮬레이션 하고자 한다. 우선 캐스케이딩을 거쳐 라우팅 맵을 구하여야 한다. 캐스케이딩을 시뮬레이션하는 것은 사실상 불가능하고 본 논문의 주제를 벗어난다. 이유는 앞서 언급한 바와 같이 Tor 네트워크의 대역폭과 과거의 패킷 밀집률로부터 통계적으로 구해지기 때문에 결과가 항상 다르다. 우리는 캐스케이딩의 결과 Fig. 6과 같은 라우팅 맵을 구했다고 가정한다. 각 라우터 사이의 숫자는 대역폭을 나타낸 것이다. A는 패킷을 보내는 소스(source) 노드이고 H는 목적지(destination)이다. 패킷은 B부터 G까지의 노드를 거쳐 H에 도달한다. H로 들어오기 직전의 노드를 찾고 그 노드로부터 H에 이르는 간선이 출구 릴레이가 되고 여기에 미러 서버를 두는 것이다.

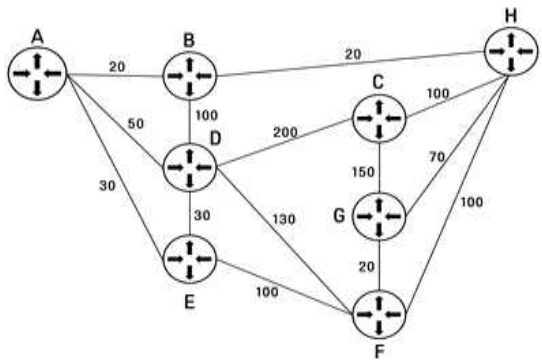

Fig. 6. Routing Map after Cascading.

Algorithm 1. Proposed Algorithm for Exit Relay Methodology

```
bool flag = false;    // global variable, to check
                      // if a packet is arrived
List exit_relays;     // exit relays on the router map

Routing(flag) {
    if (flag == true) {    //  when a packet is arrived
        Cascading(packet)
    }
    else if (flag == false) {    //  wait until packet
is arrived
        while(1) {
            if (is packet arrived?)
                flag == true;
        }
    }
    next = first relay from exit_relays;
    while (next) {
        Mirror(exit);          // check if the virus is
        next = next relay from exit_relays
    } until (next == null);
}

Cascading(packet) {     // to get an appropriate path
    router_map = Statistics(bandwidth, density);
                // build up the router map from old
bandwidth,
                    and packet density statistically
    exit_relays.add(router_map);
                // find out the exit relays from the
router_map
    return exit_relays;
}

Mirror(exit) {
    if(is virus in packet at exit) {
        Clean(virus);        //    cast out the virus
    }
    else {
        Destination(exit);  // check if the exit relay
is destination
    }
}
```

Fig. 7과 Fig. 8은 출구 릴레이를 찾는 과정을 보여준다. 패킷이 전달되는 경로를 발견하는데 있어서 가장 중요한 요소는 패킷들을 수용할 수 있는 대역폭이다. 대역폭이 중요한 이유는 Tor 네트워크의 특성 때문이다. Tor에서는 디렉토리 서버가 패킷의 흐름을 관장하는데, 일반적인 경로가 아닌 우회가 심하고 느린 독특한 경로를 선택하기 때문에 아무도 그 경로를 예측할 수 없다. 사실은 한 라우터가 경로를 선택할 때는 복호화 키와 다음 라우터의 키를 디렉토리

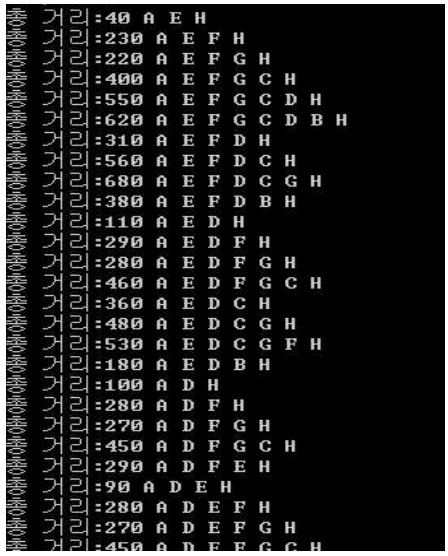

Fig. 7. Whole Path of Packets.

Fig. 8. Recommended Path of Packets.

서버로부터 받아서 다음 라우터의 경로를 결정하게 된다. 그러나 라우터가 패킷을 허용하는 이상의 대역폭을 가져야만 패킷이 지나갈 수 있다는 것은 자명하다. 그리고 우리에게 필요한 것은 출구 릴레이이기 때문에 최종 경로까지의 합을 구하였다.

우리는 100개에서 10,000개의 악성 코드를 갖는 패킷이 동시에 유입되는 것을 시뮬레이션 해보고자 한다. 깊이 우선탐색 (DFS: Depth First Search)을 통해 A 부터 H 까지 이를 수 있는 모든 경로를 구하고, 모든 경로의 대역폭의 합을 구해서 이를 총 거리라고 표현하였다. Fig. 7은 모든 경로 중 일부분을 보여준다.

Fig. 7에 는 총 거리 40에서 부터 680까지가 나타나 있는데, 이 중 500이상인 것만 Fig. 8에 모았다.

Table 2. Result of Simulation

no. of packets to enter	no. of packets to detect	percentage (%)
100	100	100
500	500	100
1,000	680	68
1,500	680	45
2,000	780	39
5,000	2,700	54
7,000	4,460	63
10,000	4,460	44

수천 개 수만 개의 패킷이 전달될 때 대역폭이 작은 경로는 선택되지 않을 것이 거의 확실하기 때문이다 [12]. Fig. 8에서 H에 직전에 나타나는 마지막 노드와 H를 연결하는 간선이 출구 릴레이가 된다. 이 중 가장 높은 빈도를 나타내는 간선을 찾아보면, F-H가 3번, C-H가 3번, G-H가 2번 나타나고, 나머지는 모두 1번 나타난다. 따라서 F-H, C-H, G-H가 모든 패킷들이 반드시 거쳐야 하는 출구 릴레이가 될 확률이 매우 높다. 여기에 미러 서버를 두고 바이러스를 검출하는 것이 가장 합리적인 것이라 판단된다.

경로 상의 대역폭은 패킷을 수용할 수 있는 양과 동일하다. Fig. 8에서 대역 폭의 합이 가장 큰 경로는 680이며 이 경로 상에서 최대 680개의 패킷을 처리할 수 있다는 뜻이다. 이 경로부터 시작해서 그 다음 넓은 대역폭을 가진 경로, 또 다음 넓은 대역폭을 가진 경로 .... 등의 순으로 차례로 패킷이 이동되게 하였다.

그 결과는 Table 2와 같다. 패킷의 수를 100개로 시작하여 10,000개까지 유입량을 늘리며 이들 중 몇 %가 12개의 경로 중 출구 릴레이가 있는 F, C, G를 거쳐 H에 도달하게 되는 지를 실험한 것이다. 예를 들어 2,000개의 패킷이 유입되었다면 680개는 G-H의 미러 서버에 의해 바이러스가 검출되고 제거된다. 다음에 B-H를 거쳐 지나가는 620개의 패킷은 미러 서버가 없기 때문에 바이러스를 검출하지 못한다. 다음에 600개의 패킷 역시 미러 서버가 없는 E-H를 거치게 된다. 나머지 100개가 대역폭 590인 경로를 거쳐 F-H 간의 미러 서버를 통해 바이러스를 검출하게 된다.

5. 결 론

뛰어난 익명성을 자랑하면서 원하는 목적지까지 완벽하게 데이터를 전송하는 Tor는 초기에는 국방용으로 만들어졌지만, 더 많이 발전되고 사용되도록 국민들에게 무료로 배포되었다. 이 결과 많은 사람들이 Tor를 이용하게 되었으나, 그것들 중에는 검열이나 검사를 피하기 위한 수단으로 사용된 것들이 많다. 다시 말하면 Tor의 가장 큰 장점인 익명성이 오히려 치명적인 오점이 된 것이다. 최근 들어 Tor의 보안을 위한 많은 방안들이 제안되고 있지만, Tor의 특성상 라우팅의 경로도 쉽게 알수 없고, Tor의 익명성을 약화시킬 수 있다는 우려 때문에 도입이 쉽지 않다.

본 논문에서는 Tor 보안 정책의 일환으로 출구 릴레이에 미러(mirror) 서버를 설치하여 패킷을 확인하고, 악성 코드를 걸러내는 방법을 다루었다. 100개부터 10,000개까지의 악성 바이러스가 담긴 패킷을 동시에 처리할 수 있는 환경을 가정하고 시뮬레이터를 만들어 실험하였고, 평균 64.1%의 검출률을 얻었다. 물론 실제의 Tor 환경에서는 실험과 같은 상황은 일어나기 힘들다. 패킷간의 처리 속도도 다르고 지연(delay)도 생기기 때문에, 소스에서 보내어진 패킷이 동시에 목적지까지 도착하는 경우는 발생하지 않는다. 또한 대역폭이 넓은 경로가 추천된다고 하더라도 반드시 그 경로를 택한다고 보장할 수는 없다. 양파 라우터의 특성상 기존의 경로가 아닌 임의의 경로를 통하여 우회할 수 있기 때문이다. 본 논문에서는 Tor의 모든 상황을 모두 고려할 수 없기 때문에 출구 릴레이를 구하는데 한정하고, 나머지 상황은 Tor에서 일어날 수 있는 최악의 상황을 가정하고 실험한 것이다.

또한 Tor의 장점인 익명성은 그대로 유지하여야 하기 때문에, 유포자의 검거나 역추적을 목적으로 해서는 안된다. 그래서 Tor에서는 패킷이 흘러가는 흐름에 미러서버를 두어 바이러스들을 채로 걸러내는 것이 유일한 대안이다. 다만 미러서버를 두는 위치를 입구에 둘 것인가, 출구에 둘 것인가, 아니면 가운데 둘 것인가 하는 문제이다. 우리는 미러서버를 출구 릴레이에 두어 Tor의 장점을 유지한 채, 좀 더 안전하게 네트워크를 이용할 수 있는 방안을 제시하였다. 랜섬웨어가 모두 Tor를 사용하여 유포되는 것은 아니기 때문에 모든 랜섬웨어를 방지할 수는 없지만, Tor를 사용하는 악성 바이러스를 탐지하는 데는 효과가 있는 방법이라 판단된다. 랜섬웨어가 기승을 부

리는 오늘날 우리가 제안한 방법이 바이러스를 방지하는 한 가지 대안이 될 수 있을 것으로 기대한다.

REFERENCE

[1] KSN, KSN Repot: Ransomeware In 2014-2016, https://securelist.com/pc-ransomware-in-2014-2016/75145 (accessed Sep., 25, 2016).

[2] The Appearance of Cryptowall Updates and New Families of Ransomeware, https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/cryptowall-updates-new-families-of-ransomware-found (accessed Sep., 28, 2016).

[3] TOR, http://torproject.org (accessed Jul., 3, 2016).

[4] What Kind of Structure is Anonymous Communication Tor, http://techholic.co.kr/archives/46127 (accessed Dec., 10, 2016).

[5] Onion Routing, https://www.onion-router.net (accessed Aug., 30, 2016).

[6] TrendMicro, *Cybercrime Hits the Unexpected, 1st Quarter Intimidation Report*, 2014.

[7] Korea Internet and Security Agency, Kr CERT/cc, *Case Analysis for the Principle of Tor Network and Relative Malicious Code,* 2014.

[8] J.Y. Seo, *A Study on Enhancing Performance and Secrecy of Tor Anonymous Communication System,* Master's Thesis of Ajou University of Technology, 2009.

[9] Red Sox, Hacking Team Traceback In-depth Analysis, https://thenewspro.org/?p=13534 (accessed Sept., 15, 2016).

[10] Akamai, *Akamai's State of the Internet/Security,* 2015.

[11] D. Easley and J. Kleniberg, *Networks, Crowds, and Markets : Reasoning about a High Connected World,* Cambridge University Press, Cambridge, U.K., 2010.

[12] M.H. Choi, B.T. Ahn, S.J. Kim, S.K. Ryu, and H.S. Kang, "An Operating Strategy of Outer Networking of University According to Traffic Efficiency Analysis," *Journal of Korea Multimedia Society*, Vol. 8, No. 1, pp. 119-127, 2005.

장 덕 성

1979년 경북대학교 컴퓨터공학과 공학사
1981년 서울대학교 계산통계학과 이학석사
1988년 서울대학교 컴퓨터공학과 공학박사

1985년~현재 계명대학교 컴퓨터공학과 교수
관심분야 : 컴파일러, 자연어처리, 음성인식 등

박 소 연

2015년~현재 계명대학교 컴퓨터공학과 재학중
2016년~현재 정보공학실험실 연구생
관심분야 : 네트워크, 컴퓨터보안, 임베디드 등

최 두 영

1994년 대구대학교 통계학과 이학사
1999년 계명대학교 컴퓨터공학과 공학석사
2000년 계명대학교 컴퓨터공학 박사 수료

현재 코리아정보기술 보안연구소 소장 재직
관심분야 : 소프트웨어공학, 컴퓨터보안, 네트워크보안 등