情報保護學會論文誌
第12券 第3號, 2002. 6

IP 역추적을 위한 새로운 접근 : 패킷 손실 기반의 논리적 전송 경로 추정

이 준 엽*, 이 승 형* 양 훈 기**, 고 재 영***, 강 철 오***, 정 주 영***

A New Framework for IP Traceback : Inference of Logical Topology by Measuring Packet Losses

J. Y. Lee*, S. H. Rhee*, H. G. Yang**, J. Y. Koh***, C. O. Kang***, J. Y. Jung***

요 약

본 논문은 다수의 호스트가 임의의 IP 주소로 목적지 호스트를 공격하는 분산 서비스거부 공격의 경우에 패킷의 전송경로를 역추적하기 위한 새로운 접근 방법에 대해 연구한다. 목적지 호스트는 전송되는 트래픽에 대하여 패킷들의 전송상태 및 전체 손실률을 계산한 후, 이를 바탕으로 소스 호스트들까지 논리적인 전송 경로를 역추적 하게 된다. 이는 동일경로를 따라 전송되는 패킷들의 손실에는 상호 연관성이 있다는 점에 근거하였으며, 시뮬레이션 결과는 특정 조건 하에서 매우 우수한 성공률을 보인다. 제안된 방식은 실시간 혹은 비 실시간으로 처리할 수 있으며, 기존의 방법들과 달리 라우터의 특정 기능이나 ISP의 도움 없이 목적지 호스트를 독자적으로 추론할 수 있다는 장점이 있다. 이 결과는 물리적인 전송경로의 추정 및 기존의 역추적 방법에 대한 보완에 적용될 수 있다.

ABSTRACT

This paper deals with study of a new framework for the traceback of distributed DoS(Denial of Service) attacks in the Internet, in which many sources flood "spoofed" IP packets towards a single victim. In our scheme, the destination host traces those anonymous packets' losses, and infers the logical end-to-end paths back towards the sources. This method is based on the fact that there is a strong correlation between packet losses when those packets traverse along a same route, and the simulation results show high probabilities of detecting the topology under a certain condition. Compared with previous approaches, our scheme has a number of distinct features: It can be performed in realtime or non-realtime, without any supports of routers or ISPs. Our results may be applied to the inference of physical topology and to support previous approaches.

Keyword : *IP Traceback, Denial of Service, Logical Topology, Packet Losses*

Ⅰ. 서 론

서비스 거부(Denial of Service : DoS)공격은 원격의 호스트나 네트워크의 자원을 소비하여 일반 사용자들이 받을 수 있는 서비스를 거부하여 서비스의 질을 떨어뜨리는 공격을 말하는데, 이는 구현이 간단한 반면 예방과 추적이 힘들어서 인터넷 보안을 위협하는 문제로 인식되고 있다.[1,2] 더구나 분산 서비스거부(Distributed Denial of Service : DDoS) 공격은 공격에 사용되는 호스트의 수가 많으므로,

* 광운대학교 전자공학부 네트워크 시스템 연구실(mjuze@explore.gwu.ac.kr)
** 광운대학교 전자공학부 다차원신호처리연구실
*** 국가보안기술연구소

공격을 받는 목적지 호스트가 이를 예방하거나 완전히 추적하여 대처하는 것은 거의 불가능한 것으로 알려져 있다. 최근에 이에 대한 IP 역추적(IP Traceback) 기술에 대한 연구가 매우 활발히 진행되어 다양한 접근 방법이 제안되고 있는데, 크게 라우터의 지원을 받는 방식[3,4]과 링크를 테스트하는 방식[1,5]으로 나눌 수 있다. 라우터의 지원을 받는 방식은 인터넷의 라우터에 특정 기능을 구현하거나 ISP(Internet Service Provider)의 도움을 받아서 전송경로를 역추적하는 방식이고, 링크를 테스트하는 방식은 역으로 트래픽을 전송하거나 ISP의 도움에 의해 경로를 추정하게 된다. 기존에 제안된 방식들은 다음과 같은 단점을 지니고 있다. 첫째, 라우터에 특정 기능을 수행할 수 있는 알고리즘을 구현하여야 하므로, 인터넷에 널리 적용되지 않은 경우에는 원하는 결과를 얻을 수 없다.[3,4] 둘째, ISP들이 자신들의 라우터에 역추적을 위한 기능을 구현하는데 소극적인 경우에 적용하기 힘들거나, 역추적을 위해 ISP의 협조를 절대적으로 필요로 한다.[3,6] [5]의 경우에는 라우터나 ISP의 지원을 필요로 하지 않으나 제안된 방법 자체가 DoS 공격이며 부정확하다.

본 논문에서는 공격을 받는 목적지 호스트가 자신에게 전송되는 트래픽을 분석하여 소스 호스트들까지의 논리적인 전송 경로를 추정하는 새로운 접근방법을 제시한다. 목적지 호스트는 패킷들을 분석하여 각각의 플로우에 대해 패킷의 손실 여부 및 전체 패킷 손실률을 측정하게 된다. 같은 시간대에 동일한 라우터를 통과하는 패킷들의 손실에는 상호 연관성이 있다는 점을 이용하면, 이에 의해 동일한 라우터를 통과한 것으로 추정되는 두 플로우를 식별할 수 있다. 이 과정을 반복하여 목적지 호스트에서 소스까지의 논리적인 전송경로를 역추적하게 된다. 이러한 원리에 의한 네트워크 토폴로지의 추론은 최근에 멀티캐스트 라우팅 트리에 적용되어 하나의 소스로부터 다수의 목적지까지의 전송경로를 추정하는데 사용되었으며,[7,8] 논문에서는 [8]에서 연구된 알고리즘을 응용하여 여러 호스트로부터의 서로 다른 패킷이 하나의 목적지로 전송되는 경우에 대해서도 적용이 가능함을 보인다.

추정된 논리적인 전송 경로는 실제의 물리적인 전송경로를 재구성하는데 적용될 수 있으며, 특히 기존의 IP 역추적 방법들이 인터넷에 일부만 구현이 되어 역추적을 위해 필요한 데이터의 일부만 얻는 경우에 추적의 성공을 위한 보조 자료로 사용될 수도 있다. 기존의 방법들과 비교하여 본 논문에서 제안된 방법이 갖는 장점은 먼저 라우터나 ISP의 도움을 필요로 하지 않는다는 점이다. 또한 공격을 받는 중에 실시간으로, 혹은 공격 후에 비 실시간으로도 적용이 가능하다는 점이다.

본 논문은 다음과 같은 내용으로 구성이 되어 있다. 2장에서는 IP 역추적을 위한 기존의 접근 방법들을 설명하고 3장에서는 패킷 손실 측정에 의해 멀티캐스트 라우팅 트리를 추정하는 방법에 대해 소개하고 IP 역추적에 응용하기 위한 방안을 설명하며, 4장에서는 제안된 방법을 사용하여 샘플 네트워크에 대해 IP 역추적을 시뮬레이션하고 그 결과를 설명한다. 5장에서 제안된 방식의 실제 적용방안 및 현재 수행되고 있는 향후 연구내용에 대해 언급하고 결론을 맺는다.

II. 기존의 방법 및 현황

2.1 기존에 제안된 IP 역추적 방식들

2.1.1 Router support 방식

Router support방식에는 다음과 같은 방식들이 있다. 우선 발신자의 소스 주소를 속이지 못하도록 하는 방법이 있는데, 이를 ingress filter라고 부른다. 이러한 ingress filter는 인가되지 않은 불법적인 발신자 주소를 차단하기 위해 라우터에 구현되며, 이러한 라우터는 합법적인 주소와 불법적인 주소를 구분할 수 있어야 한다. 따라서 트래픽이 적고 주소 체계가 상대적으로 분명한 인터넷 서비스 사업자의 경계 라우터에 구현이 가능하지만 ingress filter가 구현된 라우터는 많은 overhead를 갖게 됨으로 고속으로 패킷을 처리하는 것이 불가능해진다는 단점이 있다. 두 번째로, logging방식이 있는데, 주요 라우터에서 패킷들의 내용을 기록함으로서 패킷이 전달되는 경로를 추적하는 방식이다. 이러한 방식은 시스템에 공격이 이루어지고 난 후 그 경로를 추적하기에는 좋은 방법이지만 패킷의 표본화를 위해 많은 네트워크의 자원을 필요로 하며, 라우터들의 데이터베이스를 통합하는 것이 어렵다는 단점이 있다. 세 번째로, ICMP를 이용한 역추적 방식이 있는데 작은 값을 가지고 라우터를 지나가는 패킷들을 표본화 한 뒤 목적지 경로에 위치한 라우터에 관한 정보를 포함해서 특별한 ICMP 추적 메시지에 내용을 복사하는 방법이다. Flooding형태의

공격을 받고 이는 호스트는 ICMP 메시지의 내용을 이용해서 공격자의 경로를 재구성 할 수 있다. 하지만, 현재 구성된 네트워크 상황에서는 사용하기 복잡하다는 문제점이 있다. 마지막으로, 패킷들의 전송 경로에 위치한 라우터의 경로를 패킷에 표시하는 packet marking방식이 있다. 피해자는 패킷에 표시되어 있는 정보를 이용해서 공격자의 주소를 역추적 할 수 있으며, ISP들과의 상호 협력 없이도 구현이 가능하지만, 구현상 많은 제약이 따른다.

2.1.2 Link testing

Link testing방식에는 라우터의 input debugging 기능을 이용한 방식이 있다. 이는 특정 패킷들의 입력포트와 출력포트가 사용되고 있는지 운영자가 확인할 수 있게 해준다. Input debugging 기능을 사용할 경우 피해자는 자신이 공격을 당하고 있다는 것을 알고 있어야 하며, 공격에 사용된 패킷의 공통점을 설명할 수 있는 공격 표시(signature)를 추출 할 수 있어야 한다. 그런 다음 운영자에게 이를 통보하고, 운영자는 피해자의 출력포트에 input debugging filter를 설치한다. 이러한 필터 연결은 입력포트와 트래픽이 발생한 상위 라우터를 알려주며, 상위 라우터를 거슬러 올라가면서 반복적으로 수행된다. 다음으로 네트워크 운영자의 협조가 없이 역추적을 할 수 있는 방법 중에 많은 양의 트래픽을 역으로 발생시키고, 이러한 프래픽이 공격자로부터 전달되어 오는 트래픽에 어떤 영향을 미치는지를 측정해서 상위 경로를 추적하는 controlled flooding방식이 있다. 이는 라우터의 버퍼가 공유되기 때문에 공격자가 보낸 패킷을 포함하여 로드된 링크를 따라 이동하는 패킷은 분실 될 확률이 높다는 것에 기초했으며, 다른 링크로 옮겨가면서 최상의 소스에 도착 할 때까지 반복적으로 수행된다. 이러한 방식은 매우 정교한 방법이긴 하지만, controlled flooding방법 자체가 서비스 거부 공격이라는 것과 피해자가 상당 부분의 네트워크 토폴로지 지도를 가지고 있어야 한다는 단점이 있다. 또한 진행되고 있는 공격에는 효과적이나, 사후 분석에는 사용될 수 없다는 단점이 있다.

2.2 DDoS 도구의 공격 형태 분석

2.2.1 주요 공격 도구들[9]

DDoS 공격에 사용되는 가장 보편적인 도구들로 Trinoo와 TFN(Tribe Flood Network) 및 stacheldraht을 들 수 있다. Trinoo는 많은 소스로부터 통합된 UDP flood DoS 공격을 만들어내는 도구이다. Trinoo 공격은 몇 개의 마스터들과 그에 연결된 많은 수의 데몬들로 이루어지는데 공격자는 마스터를 통제하여 그 밑의 데몬들에게 공격 명령을 전달하게 된다. TFN은 Trinoo와 거의 유사한 분산 도구로 많은 소스에서 하나 혹은 여러 개의 목표 시스템에 대해 서비스 거부 공격을 수행한다. 또한 TFN은 UDP flood 공격뿐만 아니라 TCP SYN flood 공격, ICMP echo request 공격, ICMP 브로드캐스트 공격(smurf 공격)에 사용될 수 있으며, 공격에 사용되는 소스 IP 주소와 소스 포트는 임의로 주어지며, 패킷의 크기도 바꿀 수 있다. 마지막으로 stacheldraht는 Trinoo와 TFN의 특성을 모두 가지고 있으면서 마스터와 데몬들 사이의 통신에 암호화 기능이 추가된 것이다.

2.2.2 플로우 및 패킷 sequence의 식별

널리 알려진 DDoS 도구들 중에 TFN은 stacheldraht와 동일한 기능을 가지고 있으며, 이들 도구들은 ICMP 공격과 TCP SYN flood 공격, UDP flood 공격을 지원하는데 이들 공격에는 다음과 같은 특징들이 있다. Sstacheldraht에서 TCP SYN flood의 경우, 소스의 IP 주소와 포트 번호는 임의적으로 부여가 되고 공격에 사용되는 패킷들은 모두 동일한 sequence 번호를 가지고 있으며, 동일한 소스 IP 주소를 가진 패킷들은 동일한 포트 번호와 패킷 id 값을 가지게 된다. 또한 공격자가 보낸 패킷의 순서에 따라 주어진 범위 내에서 목적지 호스트의 포트 번호를 증가시켜가며 TCP SYN 공격을 하게 된다. 다음으로 UDP flood 공격의 경우 소스 IP 주소와 패킷 id 값은 임의적으로 변하지만, 공격에 사용된 패킷들의 소스 UDP 포트 번호와 목적지 UDP 포트 번호의 합이 일정한 값을 나타낸다. 마지막으로 ICMP 공격인 경우 같은 소스로부터 생성된 패킷들은 동일한 패킷 id 값을 가지게 된다. 이러한 DDoS 공격의 경우에 피해자 입장에서는 sequence 번호, 소스의 포트 번호, 목적지의 포트 번호 및 패킷 id 값을 분석하면 여러 소스로부터 들어오는 플로우들을 구분할 수 있으며, 각 플로우들의 패킷 sequence를 식별할 수 있다.

III. 패킷손실 측정에 의한 네트워크 토폴로지 추론

앞 절에서 분석한 바와 같이, 실제 사용되는 DDoS

공격 도구들의 경우에 목적지 호스트에서도 여러 소스로부터 플로우를 구분 할 수 있으며, 각 플로우의 패킷 sequence를 식별 할 수 있다. 이를 이용하여 본 장에서는 멀티캐스트 토폴로지 추론 방법[8]을 응용하여 목적지 호스트에서 여러 소스들까지의 논리적 전송 경로를 계산하는 방법을 논의한다.

3.1 멀티캐스트 토폴로지 추론

멀티캐스팅 트리의 루트에서 보내진 패킷의 수신 상태 및 손실률을 각 수신 노드에서 측정하면 MLE (Maximum Likelihood Estimation)에 의해 정확한 논리적 전송 경로를 구할 수 있다. 이 방법은 다음과 같이 공통 노드를 갖는 두 개의 수신노드를 식별하는 과정을 반복하여 수행된다.

$$Q \rightarrow Q' = (Q \setminus S) \cup \{S\}, \text{ with } S \subseteq Q, \#S > 1 \quad (1)$$

식 (1)에서 Q는 전체 수신노드의 집합을 나타내며, S는 Q에서 선정된 일부 수신 노드들을, 그리고 {S}는 S의 공통 부모 노드를 나타낸다. 즉, 새로운 수신 노드 집합 Q'은 Q에서 S를 뺀 나머지 수신 노드들과 이들의 공통 부모 노드로 이루어진다. 이러한 과정을 반복적으로 수행함으로써 논리적인 트리를 구성할 수 있으며, 이때 중요한 점은 공통의 부모 노드를 갖는 집합 {S}를 선택하는데 있다. 수신 노드들의 모든 조합을 분석하여 각각의 조합이 공통 노드를 가질 경우의 공통 노드의 패킷 도달률을 계산한 후, 이 중 가장 작은 값을 갖는 조합을 집합 {S}로 선정하게 되는데, 전송패킷의 수가 많아질수록 MLE에 의해 정확한 토폴로지의 재구성이 가능하게 된다.

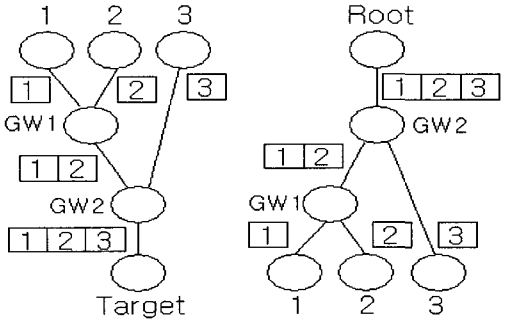
네트워크 토폴로지가 이진 트리로 구성되어 있다고 가정하면 집합 S의 선정은 다음과 같다. 임의의 수신노드 j와 k에 대해 p(j)와 p(k)를 루트에서 발생된 전체의 트래픽 중 노드 j와 k에서 수신된 패킷의 확률 값이라고 하고, r{j,k}는 동일한 패킷에 대해 두 노드 중 최소한 하나의 노드에서 패킷을 받았을 확률 값을 나타낸다. 또한 A는 이들 두 노드가 최소 공통 부모 노드를 기지고 있을 확률 값을 나타내며, A의 값이 최소인 노드들이 동일한 부모 노드를 가지게 된다. 이진 트리의 경우에 두 노드에 대한 A와 r의 값은 다음 식으로 표현된다.

$$A(\{j,k\}) = p(j) \cdot p(k) / (p(j) + p(k) - r(j,k)) \quad (2)$$

[8]에서 제안된 Binary Loss Tree(BLT) 알고리즘은 집합 Q에 하나의 노드가 남을 때까지 식 (2)를 반복적으로 적용하여 전송 경로를 구성한다. [7]에서도 유사한 방법에 의한 연구가 수행되었다.

3.2 역추적을 위한 논리적인 전송 경로의 추정

대부분의 인터넷 라우터에 구현되어 있는 Drop-tail 방식의 큐에 의해, TCP 연결들 사이에는 global synchronization이 발생하게 된다.[10] 이는 각 플로우들이 동시에 전송률을 줄이거나 증가시키는 현상을 말하는데, Drop-tail 방식이 라우터의 출력 링크에서 발생하는 각 플로우들의 패킷 손실 사이에 연관성(Dependency)을 유발하기 때문이다. 본 논문에서는 이러한 패킷손실 사이의 연관성을 이용하여, 다음과 같이 여러 소스로부터 전송된 패킷들을 분석하여 논리적 전송 경로를 추정한다..



[그림 1] (a)패킷 Flooding과 (b)멀티캐스팅

[그림 1]의 (a)에서 3개의 소스가 동시에 목적 호스트로 패킷을 전송하는 경우에 같은 라우터를 지나는 두 패킷은 패킷 손실의 상호 연관성이 매우 커서 동일한 손실률을 가진다고 가정한다. 즉, 소스1과 소스2의 패킷이 GW1을 통과하며, 각 패킷의 손실이 랜덤 변수 X와 Y를 따른다고 할 때, X와 Y는 동일한 분포를 가지고, 따라서 두 변수간의 correlation coefficient는 1로 가정한다. 두 소스의 패킷들은 GW1에서 하나의 패킷 묶음 으로 간주되어 모두 손실되거나 모두 GW2로 전송된다. 마찬가지로, 소스3의 패킷과 소스1/소스2의 패킷이 GW2를 통과할 때는 모든 패킷이 동시에 손실되거나 전송에 성공한다. 이러한 상황은 [그림 1(b)]의 변형된 멀티캐스팅 트리와 비교하여 생각할 수 있다. 루트에서 전송되는 세 개의 패킷 묶음 은 GW2로의 링크에서 동

시에 손실될 수 있으며, 전송되는 경우에는 GW1으로 가는 두 개의 패킷 묶음과 호스트3으로 가는 하나의 패킷으로 분리된다. GW1으로 가는 패킷 묶음 역시 동시에 손실되거나 나머지 두 호스트로 전송된다. 이때 [그림 1]의 (b)에서 패킷 묶음은 멀티캐스팅 트리에서 하나의 Probe 패킷에 해당하며, 따라서 공통 부모 노드의 패킷 전송률에 대한 식 (2) 역시 동일하게 적용될 수 있다.

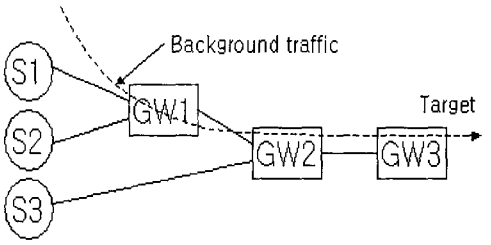
한편, GW1으로 전송되는 패킷의 손실에 따라 호스트1과 호스트2에 최소한 하나의 패킷이 전달될 확률이 작아져서 식 (2)에서 두 호스트에 대한 A(k)의 값은 최소가 되는데, 이는 [그림 1]의 (a)에 식 (2)를 적용하여도 마찬가지 결과를 얻는다. 즉, 목적지 호스트에서 패킷의 전송 상황을 분석하면, 소스1과 소스2의 패킷은 동시에 손실되거나 전송되므로 두 소스의 패킷이 하나 이상 전송될 확률은 최소가 되고, 따라서 A(k)의 값도 두 소스에 대해서 최소 값을 갖는다. 그러므로, 동일 라우터에 연결된 두개의 소스 호스트를 식별하기 위하여 3.1절에서와 마찬가지로 식 (2)를 반복적으로 수행하면, 소스 호스트에서 목적지 호스트로의 논리적인 전송 경로를 추정할 수 있다. 이 방법은 하나의 라우터를 통과하는 두 플로우의 패킷손실에 대한 correlation coefficient가 1이라는 가정을 사용하였으므로 실제의 상황과는 많은 차이가 있을 수 있다. 따라서 각 링크에서의 패킷 손실률은 사용할 수 없는 부정확한 값이나, 공통 라우터를 갖는 두 호스트의 선정은 상대적인 값을 사용하여 비교하므로 반드시 정확한 값을 사용할 필요는 없다.

Ⅳ. 시뮬레이션

4.1 UDP Flooding

본 장에서는 패킷손실 측정에 의한 IP 역추적을 위해 UDP와 TCP의 두 가지 경우에 대해 시뮬레이션 한 결과를 제시한다. 시뮬레이션은 ns[11]를 사용하였다. UDP는 sequence 번호가 없어서 손실률의 측정이 불가능하나, 트래픽 분석을 위해 ns에서 제공되는 기능을 이용하여 제안된 방식을 시뮬레이션 한다.

S1, S2, S3는 UDP CBR 트래픽을 발생시키는 소스이며, S1과 S2는 GW1, GW2를 지나 GW3로 패킷을 전달하고, S3는 GW2를 지나 GW3로 패킷



[그림 2] 시뮬레이션 구성

을 전달하게 된다. 각 패킷 크기는 210bytes이고, 0.004s 간격으로 트래픽을 발생시키며, 각 노드들은 Drop-tail queue를 사용한다. 또한 모든 링크의 대역폭은 1Mbps이며, 전송 지연 시간은 10ms이고 시뮬레이션은 10초 동안 수행되었다. 패킷의 손실 측정은 GW3에서 이루어지는데, 각 소스의 주소와 sequence 번호에 의해 각 플로우의 패킷 손실을 계산하게 된다. 다음은 target에서 수신된 각 소스들에 대한 sequence 번호이다.

[표 1] GW3에서 수신된 각 소스의 sequence 번호

S1	11111	11111	00001	11100	00011	10000	011...
S2	11111	11000	11100	00011	11000	01111	111...
S3	11111	11111	11111	11111	11111	11111	111...
Sequence	0	5	10	15	20	25	30

GW3에 기록된 정보에 기초해서 다음과 같은 값을 계산하게 된다. P[1], P[2], P[3]는 각 소스로부터 발생된 전체 트래픽 중에 수신된 패킷의 확률 값을 나타내며, r[1,2]과 r[1,3], r[2,3]는 두 소스의 조합에서 동일한 sequence 번호에 대해 최소한 하나의 패킷이 전송된 확률 값을 나타낸다. 예를 들어 r[1,2]의 경우 수신된 패킷의 sequence의 번호가 S1 :0,1,3 이고 S2 :0,1,3인 경우 r[1,2]의 값은 0.75가 된다. 또한 A(1,2), A(1,3), A(2,3)은 선택된 소스들이 공통 노드를 갖는 경우에 이 노드에서 패킷의 묶음이 성공적으로 전송될 확률, 즉 식 (2)에 의해 이 노드의 수신확률을 나타낸다. 이 시뮬레이션에서 각각의 결과 값은 다음과 같다.

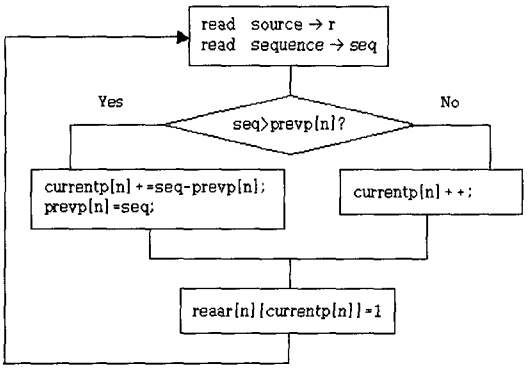
[표 2] 시뮬레이션 결과

P[1]=0.709	r[1,2]=0.859	A[1,2]=1.128
P[2]=0.402	r[1,3]=0.993	A[1,3]=1.216
P[3]=0.682	r[2,3]=0.926	A[2,3]=1.735

〔표 2〕에 의해 S1과 S2는 최소의 A값을 가지므로 공통의 노드를 통과한다고 판단하게 된다. 이러한 과정은 앞 절의 Binary Loss Tree 알고리즘에 의해 소스 노드가 하나 남을 때까지 반복적으로 수행되어 목적지 노드까지의 논리적인 전송 경로를 구성한다.

4.2 TCP Flooding

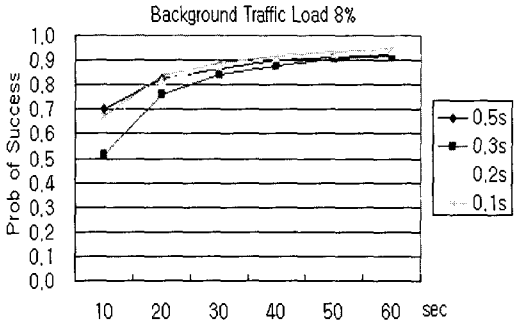
본 절에서는 TCP를 사용한 시뮬레이션을 수행한다. 네트워크 구성은 〔그림 2〕와 동일하며, S1, S2, S3는 Tahoe TCP를 사용해서 트래픽을 발생시키는 소스이고, GW1과 GW2를 지나 GW3에 도착하는 background 트래픽을 추가한다. TCP의 경우에 GW1이나 GW2에서 패킷이 폐기되면 이에 대해 각 소스들이 패킷을 재전송 한다. 이때 GW3에서는 각각의 소스에 대해 도착한 패킷의 sequence 번호가 순차적으로 증가하지 않게 된다. 이러한 점을 고려하여 패킷 손실을 측정하기 위해 다음과 같은 알고리즘을 사용하였다.



〔그림 3〕 재전송을 고려한 알고리즘

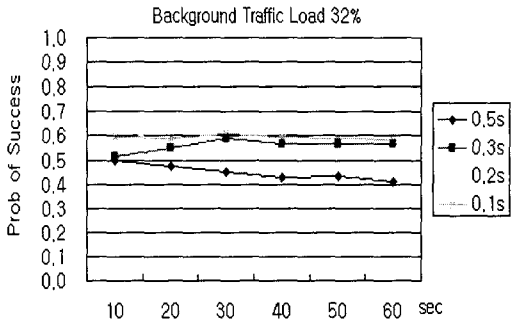
n은 소스의 수를 나타내며, 각 플로우에 대해 currentp는 현재의 수신된 sequence 번호를 나타내고, prevp는 이전의 sequence 번호를 저장한다. rearr은 각 소스로부터 수신된 패킷의 sequence 번호 증가에 따라 그 패킷이 수신되었으면 1로 표시를 하는 배열이다. 현재 도착한 패킷의 sequence 번호와 바로 이전에 도착한 sequence 번호와 비교해서 현재의 sequence의 번호가 큰 경우에는 왼쪽의 절차에 따라 sequence번호를 계산하게 되고, 현재 도착한 패킷의 sequence 번호가 이전에 도착한 패킷의 sequence번호보다 작을 경우에는 단순히 1을 증가시켜 rearr에 저장한다.

동일한 시간대에 전송된 패킷들을 분석하기 위하여, 목적지 호스트는 일정한 Time Slot 내에 도달한 패킷들에 대해 수신 상태 및 손실률을 측정하여 전송 경로를 추정한다. 즉, 작은 시간의 Slot 내에 도달한 패킷들은 Sequence 번호에 상관없이 패킷 손실에 대한 상관성이 있으므로 이에 의해 식 (2)를 적용하여 공통 노드의 추정이 가능하며, 이를 반복적으로 수행한다.



〔그림 4〕 공격시간에 따른 역추적의 성공확률

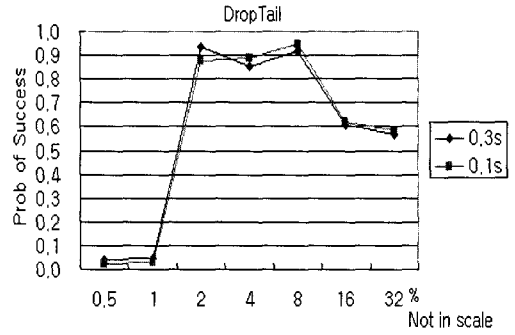
〔그림 4〕는 UDP background 트래픽이 링크 대역폭의 8%로 전송될 때 목적지 노드에서 성공적으로 전송경로를 역추적 할 확률을 나타낸다. Time Slot 값이 각각 0.1, 0.2, 0.3, 0.5sec 일 때 공통적으로 소스로부터의 공격시간이 길어질수록 성공률이 증가함을 보인다. 또한 Time Slot의 값은 큰 영향이 없음을 알 수 있다.



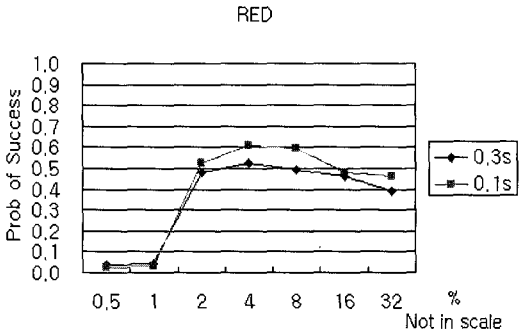
〔그림 5〕 역추적 성공확률 : Background 트래픽 32%

〔그림 5〕는 32%의 background 트래픽을 사용한 경우이다. 네트워크의 트래픽이 증가함에 따라 과도한 패킷의 손실이 발생하여 모든 Time Slot의 값에 대해 역추적의 성공률이 저하됨을 알 수 있다.

〔그림 6〕은 Time Slot 0.3 및 0.1sec에 대해



〔그림 6〕 Background 트래픽에 따른 성공률



〔그림 7〕 RED queue를 사용한 경우

background 트래픽의 증가에 따른 역추적 성공률을 나타낸다. 트래픽이 너무 작은 경우는 패킷의 손실이 없어서 성공률이 극히 낮으며, 적절한 정도의 트래픽이 있는 경우에 매우 높은 성공률을 보인다.

〔그림 7〕은 〔그림 6〕과 같은 조건 하에 GW1과 GW2에 RED queue[10]를 사용한 결과이다. RED queue를 사용하면 라우터를 통과하는 플로우들 사이의 global synchronization이 줄어들며,[10] 이는 DropTail의 경우보다 라우터를 통과하는 플로우 사이에 패킷 손실의 correlation이 작아짐을 의미한다. 따라서 제안된 방식에 의한 역추적의 정확도는 낮아진다.

V. 결론 및 향후 연구내용

본 논문에서는 목적지 호스트가 패킷의 전송상태 및 패킷 손실률을 측정하여 소스 호스트까지의 논리적인 전송 경로를 추정하는 방안을 제안함으로써, IP 역추적을 위한 새로운 방법을 시도하였다. 간단한 네트워크에서 UDP 및 TCP Flooding에 대해 시뮬레이션을 수행한 결과, 네트워크의 트래픽이 어느 정도 있는 상태에서는 90% 이상의 매우 높은 성공률을 보임을 알 수 있었다. 제안된 방식은 기존의 방법에 대해 다음과 같은 장점을 지닌다. 첫째, 공격 중에 실시간으로 혹은 공격 이후에 적용될 수 있으며, 둘째, 라우터의 특정 기능이나 ISP의 도움 없이 공격자까지의 전송 경로를 독자적으로 추적할 수 있다.

목적지 호스트가 논리적인 전송 경로를 역추적한 결과는 다음과 같은 두 가지 용도로 사용될 수 있다. 먼저, 2장에서 기술한 input debugging이나 controlled flooding과 같은 link testing을 적용하는 경우에 여러 플로우가 집중되는 노드를 우선적으로 검사하여 공격의 피해를 최소화 할 수 있다. 또한, router support 방식에서는 logging이나 marking을 지원하는 라우터의 수가 적어서 수집된 정보가 불충분한 경우에, 물리적인 전송 경로의 역추적을 수행하기 위한 보조 자료로 사용할 수 있다. 추출된 논리적 정보로부터 물리적 전송 경로를 추적할 수 있는 방법을 고안하는 경우에는 패킷 손실의 측정에 의해 독자적인 IP 역추적이 가능하다.

제안된 방식이 실제 네트워크에 적용되기 위해서는 다음과 같은 추가적인 연구가 필요하며, 이는 현재 본 논문의 저자들에 의해 연구가 수행 중이다. 먼저, 본 논문에서는 이진 트리 구조를 가진 네트워크에 대한 IP 역추적을 수행하였으며, 일반적인 토폴로지의 네트워크에 대한 연구가 필요하다. 또한 물리적인 전송 경로의 재구성 방법에 대한 연구, 인터넷 라우터에서 패킷의 손실에 대한 correlation coefficient가 1보다 작은 경우 본 논문의 알고리즘에 미치는 영향에 대한 분석, 그리고 기존의 역추적 방법에 적용할 경우의 성능에 대한 분석작업이 진행되어야 한다. 마지막으로, SYN Flooding 및 ICMP Flooding 등 실제로 적용 가능한 공격 형태를 확대할 수 있는 방안 및 역추적의 성공률을 높일 수 있는 방법에 대한 연구가 진행 중이다.

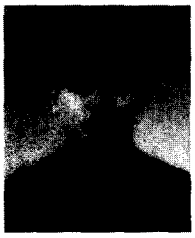
참 고 문 헌

〔1〕 Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson, "Practical network support for IP traceback," *AGM SIGCOMM*, pp. 295-306, Aug. 2000.

〔2〕 J. Scambray, S. McClure, and G. Kurtz, *Hacking exposed*, second edition, McGraw-Hill, 2000.

[3] C. Perkins, IP Mobility Support, RFC 2002, 1996.
[4] S. M. Bellovin, ICMP Traceback Messages. Internet Draft: draft-bellovin-itrace-00.txt, Mar. 2000.
[5] H. Bruch and B. Cheswick, "Tracing anonymous packets to their approximate source," Unpublished paper, Dec. 1999.
[6] J. Glave. (1998, Jan.). Smurfing Cripples ISPs. *Wired Technology News*. Available: http://www.wired.com/news
[7] S. Ratnasamy and S. McCanne, "Inference of multicast routing tree and bottleneck bandwidths using end-to-end measurements," *Proceedings of IEEE Infocom*, V.1 pp. 353~360, 1999.
[8] R. Caceres *et al.*, "Loss-based inference of multicast network topology," *Proceeding of IEEE Conference on Decision and Control*, V.3, pp. 3065-3070, Dec. 1999.
[9] D. Dittrich, papers/articles, Available: http://www.washington.edu/People/dad/
[10] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, V.1, N.4, pp. 397~413, Aug. 1993.
[11] The Network Simulator: ns-2, Available: http://www.isi.edu/nsnam/ns

〈著者紹介〉



이 준 엽 (Jun-Yeop Lee)
1995년~2001년 : 광운대학교 전파공학과(공학사)
2001년~현재 : 광운대학교 전자공학부(공학석사)
〈관심분야〉 인터넷 보안, QoS, 혼잡제어

이 승 형 (Seung-Hyong Rhee) 종신회원
1984년~1988년 : 연세대학교 전자공학과(공학사)
1988년~1990년 : 연세대학교 전자공학과(공학석사)
1995년~1999년 : University of Texas at Austin, Dept. of ECE(Ph. D.)
1990년~1995년 : 국방과학연구소 연구원
1999년~2000년 : 삼성종합기술원 전문연구원
2000년~현재 : 광운대학교 전자공학부 조교수
〈관심분야〉 인터넷 보안, QoS, 혼잡제어, 트래픽관리

양 훈 기 (Hoon-Gee Yang)
1981년~1985년 : 연세대학교 전자공학과(공학사)
1985년~1987년 : SUNY at Buffalo(공학석사)
1988년~1992년 : SUNY at Buffalo(Ph. D.)
1993년~1996년 : 광운대학교 조교수
1997년~현재 : 광운대학교 정교수
〈관심분야〉 신호처리, 암호이론, 이동통신

고 재 영 (Jae-Young Koh)
1980년~1984년 : 전북대학교 전자공학과(공학사)
1990년~1992년 : 전북대학교 전자공학과(공학석사)
1993년~1998년 : 전북대학교 전자공학과(공학박사)
1984년~2000년 : 국방과학연구소 선임연구원 팀장
2000년~현재 : 국가보안기술연구소 책임연구원 부장
〈관심분야〉 네트워크 보안, VPN, 방화벽 시스템

강 철 오 (Cheol-Oh Kang)
1989년~1993년 : 인하대학교 전자계산학과(공학사)
1983년~1985년 : 인하대학교 전자계산학과(공학석사)
1995년~1999년 : 국방과학연구소 연구원
2000년~현재 : 국가보안기술연구소 선임연구원
〈관심분야〉 네트워크/시스템 보안, 통합보안관리, 은닉 채널

정 주 영 (Ju-Young Jung)
1993년~1996년 : 중앙대학교
1997년~2001년 : 한국외대학교 컴퓨터공학과(공학사)
2001년~현재 : 숭실대학교 대학원 컴퓨터공학과(공학석사)
2001년~현재 : 국가보안기술연구소
관심분야 : 네트워크 보안, 클러스터링, 3D Game 엔진