Trust 기반의 DoS 공격에 대한 False-Positive 감소 기법

박종경°, 이태근, 강용혁, 엄영익
성균관대학교 정보통신공학부
e-mail: {jparkk°, tedlee, yhkang1, yieom}@ece.skku.ac.kr

Trust Based False-Positive Reduction Scheme against DoS Attacks

Jong Kyung Park°, Taekeun Lee, Yong-hyeog Kang, Young Ik Eom
School of Information and Communication Engineering,
Sungkyunkwan University

요 약

최근의 네트워크 공격의 주류는 DoS (denial-of-service)와 DDoS (distributed DoS) 공격이다. 이러한 공격들은 공격자가 침입 대상 시스템의 자원을 완전히 소모시켜서 시스템이 정상적인 서비스를 할 수 없도록 하는 것이다. 각 시스템의 관리자들은 이러한 침입이나 공격을 막기 위한 방편 중에 하나로 IDS(intrusion detection system)를 사용하고 있다. 그러나 IDS의 높은 false-positive(정상적인 사용을 공격으로 잘못 판단하는 경우)의 발생빈도는 심각한 문제점 중의 하나는 이다. 이런 false-positive의 발생빈도를 줄이고자 본 논문에서는 한번의 판단만으로 연결(connection)을 차단시키지 않고, trust라는 개념을 도입하여 trust의 값에 따라서 사용자에게 차등 서비스를 제공하는 기법을 제안한다. 즉, trust를 이용하는 기법은 각 사용자를 한번에 공격자인지 일반 사용자인지 결정하지 않고, 한 번 더 검사하여 false-positive의 발생빈도를 감소시키는 기법이다.

1. 서론

DoS(Denial-of-Service) 공격은 한 사용자가 시스템의 자원을 독점하거나 모두 사용, 또는 파괴함으로서 다른 사용자들이 이 시스템의 서비스를 올바르게 사용할 수 없도록 만드는 것을 말한다. 최근에는 DoS 공격 또는 DDoS(Distributed Dos) 공격이 네트워크 공격의 주류를 이루고 있다. 이러한 공격들을 방어하기 위해서 많은 시스템 관리자들은 IDS(Intrusion Detection System)를 사용하고 있다. 공격자를 막기 위한 시스템인 IDS의 높은 false-positive 발생 빈도는 심각한 문제점 중에 하나이다. False-positive라는 것은 IDS가 사용자의 정상적인 패킷을 공격으로 잘못 판단하는 것을 말한다. False-positive의 발생 빈도가 높게 되면 그 IDS의 성능은 떨어지게 된다.

이러한 IDS의 문제점을 보안하기 위해서 본 논문에서는 CBQ(Class Based Queuing)를 이용한다. 우선 각 사용자의 패킷들을 trust값으로 평가하여 등급별로 나누고, 그 나누어진 등급에 따라서 서비스를 함으로써 false-positive의 발생빈도를 감소시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해서 소개하고, 3장에서는 trust의 기준과 그 CBQ 사용의 메커니즘에 대해서 설명한다. 4장에서는 결론 및 향후 과제에 대해서 기술한다.

2. 관련연구

본 장에서는 CBQ의 기본 개념과 패킷의 흐름 상태를 바탕으로 대역폭을 나누어서 공격에 제한을 주는 메커니즘을 소개한다.

본 연구는 대학 IT연구센터 육성 지원사업의 연구결과로 수행되었음.

2.1 CBQ(Class Based Queuing)의 기본개념

CBQ는 우선순위 큐잉(priority queuing) 방식의 변형으로서 하나의 출력 큐 대신에 여러 개의 출력 큐를 클래스별로 두어서 우선순위를 정하고 각 큐별로 서비스 되는 트래픽의 양을 조절할 수 있는 큐잉 기법이다[1].

CBQ의 구성 요소는 packet classifier, link-sharing framework, packet scheduler, estimator 그리고, management interface로 구성되어있다. 각 구성요소의 기능은 다음과 같다.

- packet classifier : 각 패킷들을 클래스별로 분류한다.
- link-sharing framework : Interface에 대해서 link-sharing에 제한을 유지한다.
- packet scheduler : Class들의 대역폭이나 우선순위에 따라서 각 class들을 스케쥴링 한다.
- management interface : Class들을 생성하거나 삭제하는 역할을 한다.
- estimator : 각 class가 할당된 대역폭을 넘는지 아닌지 검사한다.

2.2 CBQ and Traffic Monitoring 메커니즘

Traffic Monitoring 메커니즘은 트래픽을 모니터링 하여 DoS 공격을 찾아내거나, 어떤 사용자가 많은 대역폭을 사용하는 경우 그것에 적합한 대역폭을 갖고 있는 큐를 할당하는 메커니즘이다[2].

메커니즘은 3개의 스레드(thread)로 구성되어 있다. 첫 번째 스레드는 패킷의 목적지 주소(destination address)와 출발지 주소(source address)를 모니터링 한다. 두 번째 스레드는 어떤 출발지 IP 주소가 많은 패킷을 송수신하는지 검사한다. 세 번째 스레드는 시스템 관리자로부터 명령들을 받을 수 있도록 대기한다.

첫 번째 스레드에서 출발지 IP 주소를 기준으로 각각의 사용자들을 구분한다. 이렇게 구분된 사용자들을 두 번째 스레드가 감시하게 된다. 공격이 의심되거나 공격으로 판단되는 사용자를 세 번째 스레드가 시스템 관리자로부터 명령을 받아서 처리하게 된다.

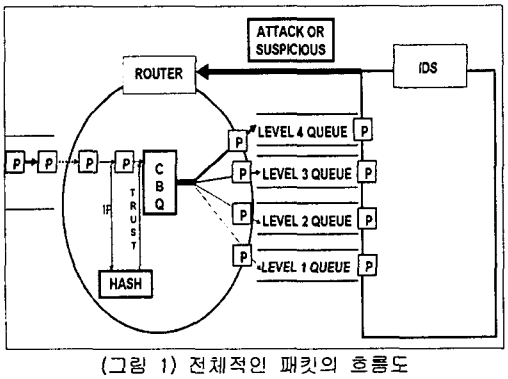
패킷들은 4개의 클래스로 분류가 된다. 클래스는 총 4개로 구성되며 그 기준은 다음과 같다.

1. Class 1 : 작은 패킷들을 매우 많이 전송하는 경우에 해당된다. 이것은 DoS 공격으로 여긴다. 이러한 경우는 해당 출발지 IP 주소에서 오는 패킷들을 버린다.
2. Class 2 : 장시간 동안 많은 대역폭을 사용하는 경우에 해당된다. 이런 경우의 처리방법은 큐를 한 단계 아래의 경우로 내린다.
3. Class 3 : 짧은 시간 많은 대역폭을 사용하는 경우이다. 이러한 경우는 일반 사용자의 경우이다. 이런 경우는 일정 기간동안 사용 시간을 검사하여 기준을 넘길 경우는 클래스 2로 이동 시킨다.
4. Class 4 : 많은 대역폭을 사용하지는 않지만 매우 많은 패킷을 발생시키는 경우는 시간을 검사하여 일정시간 이상 계속 이루어질 경우에는 해당 출발지 IP 주소로부터 오는 패킷들을 버린다.

4개의 클래스로 분류된 패킷들은 서로의 클래스로 보내지거나 버려짐으로서 관리된다.

3. Trust를 이용한 false-positive 감소 기법

3장에서는 시스템의 구성요소, trust의 정의,trust의 기준, trust의 관리 방법과 큐의 구성과 그 기준, 전체적인 동작 메커니즘에 대해서 설명한다. 본 논문에서 제안하는 기법은 그림 1에서 보이는 바와 같이 라우터와 라우터에서 동작하는 CBQ, hash table, 그리고 IDS로 구성이 된다. IDS에서 공격이나 공격이 의심스러운 상황을 검사하게 되면 이러한 정보를 라우터에 전달하여 라우터에서 사용자에게 제한을 가한다.



(그림 1) 전체적인 패킷의 흐름도

3.1 Trust의 정의와 기준

본 3.1 장에서는 trust의 정의와 trust의 값의 변경되는 방식에 대해서 설명하겠다.

Trust의 정의는 공격자와 일반 사용자의 판단을 하는데 있어서 사용자들을 단계적으로 구분하는 개념으로. 네트워크 사용자들에 대한 일종의 신용도이다. Trust의 값이 변경되는 방식은 다음과 같이 이루어진다.

- Trust 값은 1~4사이의 값을 갖는다.
- 기본적으로 각 사용자의 초기 trust 값은 최 상위 값(4)을 갖는다.
- IDS에서 의심(suspicious:많은 트래픽을 유발하여 공격이 의심스러운 단계)단계를 발견하여 신호(signal)를 라우터로 전달했을 경우 trust값을 1 감소시킨다.
- IDS에서 공격을 발견하여 신호를 라우터로 전달했을 경우 trust값을 2 감소시킨다.
- 일정 기간 동안 공격이나 의심 상태가 없을 경우에는 trust 값을 1 증가 시킨다.
- Trust 값이 1 미만으로 내려갔을 경우에는 해당 사용자의 패킷을 버린다.

3.2 큐(Queue)의 구성

큐는 총 4개로 구성이 된다. 각 큐의 세부 사항은 다음과 같다.

1. 첫 번째 큐는 일반 큐로서 전체 대역폭의 50%를 차지한다. Trust 값은 4를 갖는다.
2. 두 번째 큐는 의심상태가 된 단계로 전체 대역폭의 30%를 할당 받게 된다. Trust 값은 3이 된다.
3. 세 번째 큐는 전체 대역폭의 15%를 할당 받게 된다. Trust 값은 2가 된다.
4. 마지막 큐는 가장 낮은 trust의 값을 갖는 단계로 전체 대역폭의 5%를 차지한다. Trust 값은 1이 된다.

본 논문에서 제안하는 기법에서 큐의 구성 기준은 다음과 같다.

1. 공격이 결정되는 단계는 일반적으로 의심단계에서 계속 패킷들이 공격적인 패턴으로 전송될 경우 공격으로 판단한다. 따라서 공격이 결정될 경우 trust 값은 의심(1)단계+공격(2)단계해서 총 3이 감소한다. 따라서 이러한 사용자의 false-positive를 피하기 위해서는 trust값이 3 이상이 필요하게 된다.
2. 3 단계 이상일 경우 한 사용자가 일반적인 사용자이지만 일시적으로 짧은 기간에 여러 번 트래픽을 많이 발생시켜서 바로 차단되는 경우도 방지 할 수 있다.
3. 의심 단계의 필요성은 공평성(fairness)때문이다. 한 사용자가 많은 대역폭을 사용할 경우 다른 사용자들에게 피해를 준다. 따라서 본 메커니즘에서는 대역폭을 많이 사용하는 사용자들을 의심 단계라는 한 레벨 낮은 단계로 내림으로서 공평성을 얻을 수 있다.

이러한 근거로 종합적으로 판단하여 총 4단계의 큐로 구성 하였다.

3.3 Trust의 관리

Trust는 각 사용자의 출발지 IP 주소를 hash table을 사용하여 저장한다. 각 출발지 IP 주소에는 trust 값과, 마지막으로 trust의 값이 변경된 시간(Time)이 저장 되어 있다. 그 구조는 그림 2에서 보이는 바와 같다.

```
typedef struct{
        Int Trust;
        Int Time;
}
```

(그림 2) Trust의 구조

Hash의 값이 같을 경우에는 linked-list로 관리한다.

3.4 동작 메커니즘

패킷이 라우터에 입력되면 hash table에서 해당 패킷들의 trust 값을 찾게 된다. 찾은 trust 값에 의해서 패킷은 자신에게 맞는 큐를 통해서 IDS로 전송된다. IDS에 수신된 패킷은 IDS에 의해서 공격여부를 판단 받게 되고 그 결정은 다시 trust 값에 영향을 주게 되어서 다음에 입력되는 패킷들의 향방을 결정한다. 동작 메커니즘은 총 3개의 처리과정으로 구성되며 라우터에서 수행된다.

첫 번째 처리과정은 알고리즘 1에서 보이는 바와 같다.

```
Receive (packet [i])
addr[i] <- packet source address
Trust[i] <- hashF(addr[i])
if( Trust[i] < Trust_threshold(1))
        Then packet drop
switch (Trust[i])
        Case 4: Enqueue (level 4, packet[i])
        Case 3: Enqueue (level 3, packet[i])
        Case 2: Enqueue (level 2, packet[i])
        Case 1: Enqueue (level 1, packet[i])
```

(알고리즘 1) 처리과정1의 수행 과정

첫 번째 처리과정은 라우터에 입력된 패킷의 출발지 IP 주소를 가지고 hash table을 검색하여 trust의 값을 찾아내고, 찾아낸 trust 값에 적합한 큐로 패킷을 보낸다.

처리과정 2는 공격이나 의심상태 여부에 따라서 사용자의 trust 값을 변경 시켜준다. 처리과정 2는 알고리즘 2에서 보이는 바와 같다.

```
Listen signal (addr[i]) // from IDS
if (signal=attack)
  Then Trust[i]<-Trust[i]-2
       Time <- nowT // nowT 현시간
if (signal=suspicious)
        Then Trust[i]<-Trust[i]-1
                Time <- nowT
```

(알고리즘 2) 처리과정 2의 수행 과정

두 번째 처리과정은 IDS로부터 신호를 기다리고 있다가 공격신호가 오면 trust 값을 2 감소시키고 Time은 현재 시간으로 조정한다. 의심 신호가 오면 trust 값을 1 낮추어 주고 Time은 현재 시간으로 조정한다.

```
if ((nowT-Time) > Time_threshold)
        Then Trust[i]<-Trust[i]+1
```

(알고리즘 3) 처리과정 3의 수행 과정

세 번째 처리과정은 알고리즘 3에서 보이는 바와 같다.

세 번째 처리과정은 hash table을 주기적으로 검사하여 Time 값이 일정 시간 이상인 경우는 해당 출발지 IP 주소에 대해서 trust 값을 1 증가 시켜준다.

4. 결론 및 향후 연구

IDS는 보안 분야에서 가장 중요한 요소 중에 하나이다. 하지만 이러한 IDS에서 가장 심각한 문제점 중의 하나는 바로 false-positive이다. False-positive의 발생빈도가 높으면 IDS의 신뢰성과 성능에 나쁜 영향을 준다. 이러한 문제점을 줄이기 위해서 본 논문에서는 출발지 IP 주소마다 trust 값을 부여함으로서 "공격자" 아니면 "일반 사용자"라는 이분법적인 구분이 아니라 각 출발지 IP 주소의 trust 값에 따라서 사용자들에게 차등적인 서비스를 제공하는 기법을 제안하였다. 이러한 기법을 사용함으로서 각 사용자의 패킷들은 공격여부를 판단 받을 때 한 번 더 검사받게 된다. 따라서 false-positive의 발생빈도가 감소하게 된다. 또한 의심이라는 단계를 두어서 많은 대역폭을 사용하는 사용자들에게는 일종의 패널티(대역폭을 제한)를 부여함으로서 공평성을 얻는다.

향후 연구에서는 본 논문에서 제시한 기법을 시뮬레이션을 통하여 검증하며, 테스트 결과를 이용해서 시스템에 공격이 발견되지 않아서 trust를 증가 시키는 기준값인 time_threshold도 연구 할 것이다.

참고문헌

[1] Sally Floyd and Michael Francis Speer, "Experimental Results for class-based Queueing," November 11,1998.

[2] Frank Karg, Joern Maier, and Michael Weber, "Protecting Web Servers from Distributed Denial of Service Attacks,"

[3] Ian Wakeman, Atanu Ghosh, and Jon Crowcroft, "Implementing Real Time Packet Forwarding Policies using Streams," November 14. 1994.

[4] 고광선, 강용혁, 엄영익, "리눅스시스템에서 서비스 자원소비율을 이용한 분산서비스거부공격 탐지 기법," 한국정보처리학회 2003 춘계 학술발표논문집, Vol. 10, No. 1, May. 2003, pp. 2041-2044.

[5] Ian Wakeman, Atanu Ghosh, and Jon Crowcroft*, "Implementing Real Time Packet Forwarding Polices using Streams,".