IP 주소 프로파일링과 Pi 마킹을 이용한 DDoS 공격 대응

아주대학교 정형철 · 홍만표*

1. 서 론

인터넷 인프라의 급속한 양적 팽창 및 비약적인 기술 발전에 따라 사회 전반에 걸친 기반들이 인터넷에 의존하고 있는 상황에서 분산 서비스 거부 공격(DDoS: Denial of Service, 이하 DDoS 공격)은 인터넷 인프라의 매우 큰 위협이 되고 있다. 일례로 2000년 초 야후(yahoo), 이베이(E-bay) 등의 웹 서버등이 서비스 거부 공격을 받는 사건이 발생했으며, 2002년 10월, 인터넷의 핵심 기능을 수행하는 최상위 도메인 네임 서버 13개가 서비스 거부 공격을 받은 바 있다[21]. 인터넷 인프라에 대한 공격은 진화를 거듭하며 2003년 1월 25일, 신종 웜 바이러스인 슬래머(slammer)는 미국, 캐나다 등 전 세계적으로 확산되면서 인터넷 전체를 마비시켰다. 슬래머는 취약성을 가진 MS-SQL 서버에 침투하기 시작하면서, 같은 취약성을 갖는 또 다른 MS-SQL 서버를 탐색하며 급속도로 확산되었다. 감염시도 자체가 폭발적인 트래픽을 유발함으로써 도메인 네임 서버 등이 제공하는 인터넷 서비스 전반에 걸쳐 접속 지연 및 불가 상태가 발생하였다.

DDoS는 서비스 거부 공격(DoS: Denial of Service, 이하 DoS 공격)을 기원으로 하며, DoS 공격은 특정 인터넷 서비스를 이용하는 사용자들이 원하는 서비스를 이용하지 못하도록 해당 서버를 가동 불능 상태로 만드는 모든 공격 행위를 칭한다[5]. DoS 공격은 공격 대상 서버의 설계상 또는 구현상의 취약점을 이용하여 해당 서버를 직접 공격하는 직접 공격과 근접 지역의 네트워크에 부하를 일으켜 트래픽 정체를 유발하는 간접 공격으로 나누어 볼 수 있으며, 최근에 들어서는 간접 공격이 늘어나고 있는 실정이다. 이는 공격하고자 하는 대상 서버가 특별한 취약점이 없는 상황에서도 공격 당할 수 밖에 없다는 이유 때문에 더욱 대책이 시급하다.

DDoS 공격은 DoS 공격의 진화된 형태이며, 불특정 다수의 호스트에 DoS 공격 도구들을 설치하여 공격시 이를 이용한다. 이런 호스트들을 좀비(zombie)라고 하며, 그림 1에서 보듯이 설치된 좀비들은 공격자의 명령에 따라 공격 대상에 DoS 공격을 일으켜 대상 서버를 마비시킨다. 향후 등장 가능한 DDoS 공격은 슬래머의 경우와 같이 좀비의 설치 자체가 바로 공격이 될 수 있다.

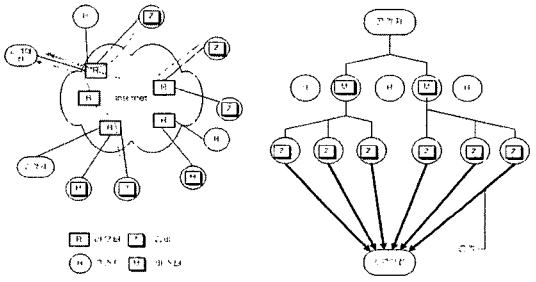


그림 1 DDoS 공격의 형태. 좌측 그림은 DDoS 공격이 인터넷에 물리적으로 존재하는 모습을 보여주며, 우측은 DDoS 공격의 계층적인 논리 구조를 보여준다.

DDoS 공격에 대한 여러 가지 대응 방법이 제안되었으며 그 중 하나는 패킷의 내용을 검사한 후 공격 트래픽을 추출하는 방법이 있었다[22]. 이 방법은 TCP SYN 공격이나 UDP 범람과 같은 특정 유형의 공격에 대한 필터링 규칙을 가지고 해당 조건을 만족하는 패킷을 필터링하는데, DDoS 진화에 따라 새로운 공격 형태가 등장할 경우 즉각적으로 대응할 수 없다는 단점을 가지고 있다. 또 다른 방법은 공격이

* 종신회원

진행되는 동안 트래픽을 분석하여 근원지를 찾거나 공격 트래픽을 추출하는 방법을 사용하였는데, 이는 대처 방법을 찾기까지 DDoS 공격에 무방비 상태가 되므로 적합한 대응 방법이 될 수 없다. 현대의 산업 기반 중 많은 부분이 인터넷에 의존하고 있는 상황에서 DDoS 공격에 신속히 대응하지 못함은 치명적인 재정적 손실을 야기할 수 있다.

결과적으로 향후 다양한 DDoS 공격에 적응 가능하며 공격 발생과 동시에 대응 가능한 시스템이 요구된다. 본 논문에서는 DDoS 공격의 변화에 면역력을 가지며, 공격이 이루어지는 시점과 동시에 대응 가능한 방법을 제안한다. 제안된 시스템은 공격 발생 전 필터링 정보 즉, 정상 호스트를 판단하는 단계와 공격 발생시 정상 호스트의 트래픽을 보호하는 단계로 구성된다. 정상 호스트의 트래픽을 보호하는 데 초점을 맞춤으로써 DDoS 공격의 변화에 적응력을 가지며, 미리 생성된 필터링 정보를 이용함으로써 공격 발생시 추가적인 작업을 하지 않고 즉각적으로 대응할 수 있을 것이다.

본 논문의 구조는 다음과 같다. 제 2장에서는 관련 연구에 대해 살펴보고 제 3장에서는 이를 보완할 대응 시스템을 제안한다. 4장에서는 실험 결과를 보여주고 5장에서 결론을 내린다

2. 관련 연구

2.1 기존 대응 방법

DDoS 대응 방법은 여러 가지 분류 기준으로 나누어 볼 수 있다. 시점에 따라 분류하면, 간략히 공격 발생 전과 후로 나눌 수 있다. 공격 발생 전의 대응은 보호 대상인 시스템 또는 네트워크의 취약점을 보완하는 작업이다. 또는 인터넷 인프라를 개선하여 DDoS에 대한 적응력을 높일 수도 있다[19,25]. 공격 발생 후 대응은 DDoS 공격으로 인한 트래픽을 처리하는 과정이다. 필터링, 전송율 제한[14,22,25] 등으로 공격 트래픽을 제거 또는 제한하며, 공격 트래픽 식별 방법이 매우 중요한 관건이다. 또한 공격이 시작되는 실제 위치를 색출하기 위해 IP 트레이스백(trace back)[23] 기법이 사용되었는데 이는 대량의 공격 패킷을 수용한 후에야 공격 근원지를 파악할 수 있다는 단점이 있다. 대응 지점에 따라 보면, 공격이 시작되는 지점[10], 중간 경로[16,17,23,24], 공격 대상 지점[13,25]으로 각각 구분된다. 공격이 시작되는 위치에서의 대응 방법은 공격 트래픽들이 네트워크로 유입되기 전에 차단할 수 있다는 장점이 있지만, 공격 시작 지점에서는 정보 수집의 어려움 때문에 공격 트래픽을 효과적으로 구분하기 어렵다. 중간 경로상의 대응은 각 라우터들이 분담하여 공격 트래픽을 처리한다. 위조된 출발지 주소를 가진 패킷을 실제 라우터 경로와 비교하여 구분하는 방법[16,17]과 공격 대상 지점에서 보내온 필터링 정보를 라우터간 공유를 통하여 대응[25]하는 방법이 있었다. 이 방법은 라우터에 해당 대응 시스템이 모두 설치되어야 효과를 발휘할 수 있다. 공격 대상 지점에서는 공격 패킷이 실제 연결을 확립할 수 없음을 이용하여 웹 서버를 보호하거나[13], 사용 빈도를 통해 정상 호스트를 선별하여 서버를 보호하는 방법이 있었다[26].

현재 인터넷 인프라에서 DDoS 공격을 막기 어려운 이유는 첫째, 공격 트래픽과 정상 트래픽의 구분이 모호하기 때문이다. 공격 트래픽은 대량의 TCP SYN 패킷이거나 UDP 패킷등인데, 현 인터넷은 정상적인 서비스를 위한 패킷인지 아니면, 공격 의도를

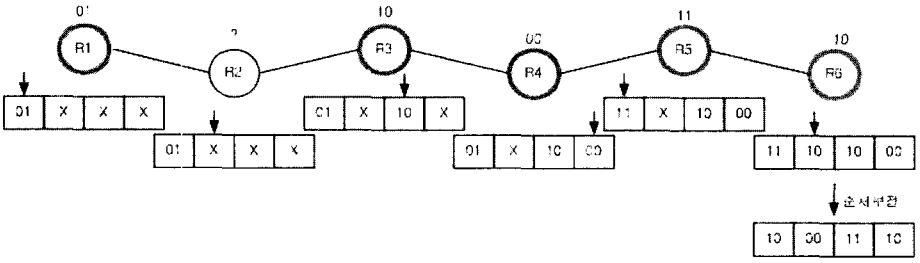


그림 2 Pi 값을 얻는 과정. 위 그림은 각 라우터 IP 주소 하위 2비트를 마킹값으로 하고 있다. 이해하기 쉽도록 Pi 필드를 8비트로 표시하였다. 실제의 경우 16비트가 사용된다.

가진 패킷인지를 구분할 수 있는 장치가 마련되어 있지 않다.

둘째, 공격 트래픽 대부분은 출발지 주소가 위조되어 있다. 공격이 실제 시작되는 위치를 판단할 수 없는 이유이다. Ingress 필터링, IP 트레이스백(traceback) 등의 기법이 나와 있지만, 전자는 모든 게이트웨이에 설치되어야 한다는 가정이 있으며, 후자는 대량의 공격 패킷을 받고 난 후에 근원지를 알 수 있다는 단점이 있다.

위 두가지 문제점을 해결하기 위한 연구 중 첫 번째 문제에 대해 히스토리 기반 IP 필터링을 두 번째 문제에 대해 Pi의 장점과 단점을 검토한다.

2.2 히스토리 기반 IP 필터링

히스토리 기반 IP 필터링[26]은 공격 중 트래픽 가운데 공격 트래픽을 색출하는 방법이 아니다. 이와는 반대로, 정상 호스트를 선별하고 이 정상 호스트로부터 오는 트래픽을 제외한 나머지를 공격이라 가정하고 제거 또는 제한한다. 정상 호스트를 선출하는 기준은 호스트의 접근 빈도이다. 접근 빈도란 두 가지로 정의되었는데, 하나는 시간당 접근 패킷수, 또 하나는 시간당 접근 데이터량으로 정의되었다. 접근 빈도가 높은 IP 중 약 82.9%는 이전에도 여전히 서버를 이용하였다는 것을 발견한 연구가 있었다[9]. 따라서, DDoS 공격시 히스토리 기반 IP 필터링을 사용하면 최소한 82.9%의 사용자는 공격으로부터 보호할 수 있다고 할 수 있다. 이 제안 방법에는 한 가지 문제점이 있는데 단순히 접근 빈도만을 가지고 정상 호스트를 선출하기 때문에 이 대응 시스템을 알고 있는 공격자는 임의로 출발지 주소가 위조된 패킷을 대량으로 생성하여 이 시스템의 정상 IP에 등록시킬 수 있다. 따라서, 정상 IP 선별 과정에 더욱 엄격한 방법이 요구된다.

2.3 Pi(Path Identification)

DDoS 공격을 차단하기 어려운 한 가지 이유는 출발지 IP 주소가 위조되어 있기 때문이다. 따라서 공격 패킷의 근원지를 확인하고 이에 대한 대응을 하기 매우 어렵다. Abraha Yaar는 Pi(경로 식별자: Path Identification)[1]를 사용하여 출발지 IP 주소 위조에 상관없이 패킷의 지나온 경로를 간접적으로 추론하는 방법을 제안하였다.

Pi에서는 각각의 라우터 IP 주소 하위 n-비트를 각 라우터의 식별자로 간주하며, 라우터는 자신을 거쳐가는 패킷의 특정 필드에 자신의 식별자를 기록한다. 기록되는 위치는 현재 사용되지 않고 있는 프래그먼트 필드이며, 최대 16비트까지 사용이 가능하다.

그림 2에서 보듯이 라우터 R1으로부터 라우터 R6까지 도달한 패킷은 경로 식별자 10, 00, 11, 10의 값을 가지게 된다. 그림에서 ?로 표시된 부분은 Pi 마킹을 지원하지 않은 라우터이며, 라우터의 식별자로 2비트를 사용하였고 실제 사용되는 필드는 16비트이지만, 간략히 8비트를 사용하는 예를 들었다.

DDoS 공격이 발생하면 각 Pi 값마다 패킷의 수를 합산하며, 일정 수 이상의 값을 가지는 Pi 값을 공격 경로의 Pi 값으로 인식하고 해당하는 패킷을 필터링함으로써 공격에 대응한다.

Pi 대응 방법은 Pi 값이 특정 값에 편중되어 있다면 정상 패킷과 공격 패킷간의 구분이 모호해진다는 단점을 가지고 있다. 예를 들어 공격 대상 지점으로 가는 경로에서 추출한 Pi 값이 대부분 $m \cdot ln(m)$ 개 값에 집중되어 있다면, 공격자가 개 이상의 좀비를 가진다면 모든 패킷을 공격 패킷으로 간주하게 되며 정상적인 호스트에게 서비스를 제공할 수 없다.

3. 제안된 DDoS 대응 시스템

3.1 개요

본 논문에서 대상으로 하는 DDoS 공격은 공격 대상이 되는 특정 호스트 - 보호 대상 서버 혹은 공격 대상 서버 - 로 대량의 패킷을 전송하여 서버의 자원 고갈이나 주위 네트워크를 마비시키는 공격으로 한정한다. 이 경우 공격자가 관장하는 좀비들이 설치된 호스트를 공격 호스트라 하고, 정상적인 서비스를 요청 및 수행하는 호스트를 정상 호스트라고 정의한다. 정상 호스트가 보내는 트래픽을 보호하는 것이 최종 목표이며, 2단계의 과정으로 나누어 수행된다. 1단계에서 보호 대상 서버의 정상 호스트를 결정한다. 접근 빈도와 사용 패턴이 정상적인 호스트를 선정하며 각 호스트는 IP 주소로 구분된다. 이 정상 호스트를 식별하기 위한 방법은 네 가지로 요약되며 뒷절에서 설명한다. 2단계에서는 정상 호스트의 패킷들을 제외한 나머지 패킷의 전송율을 제한하여 정상 트래픽이 원활히 소통될 수 있게 한다. 정상 호스트를 식별

하는 방법에 따라서 식별 목록의 크기나 효과가 차이가 나며, 식별 목록의 크기를 줄이면서 효과가 높은 방법을 찾는 것이 최우선 과제이다.

DDoS 공격 대응 시나리오는 그림 3에서 보듯이, 첫째 보호 대상 서버는 사용 빈도가 높고 사용 패턴에 부합되는 정상적인 호스트를 선정한 후 이들의 목록을 생성한다. 공격 발생시 미리 생성된 이 목록을 경계 라우터에게 전달하고 경계 라우터는 이 정보를 바탕으로 정상 호스트가 보낸 패킷은 통과시키고 이외의 패킷은 전송율을 제한하거나 제거한다.

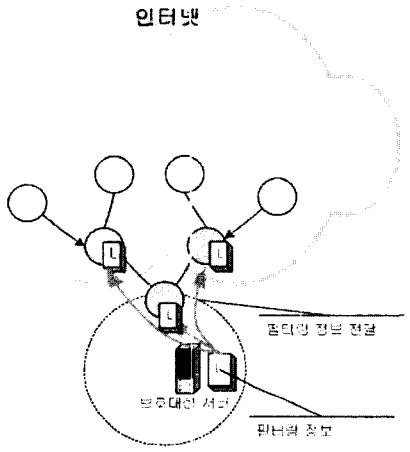


그림 3 DDoS 공격 대응 구조도. 보호 대상 서버는 필터링 정보로 보호 대상 호스트 IP 주소 목록을 생성하고 공격 발생시 경계 라우터에게 필터링 정보를 전달한다.

3.2 정상 호스트 선별

정상 호스트 선별은 보호 대상 서버에 높은 접근 빈도를 가지며 사용 패턴이 해당 서버에 부합되는 호스트를 선별하는 과정이다. 각 호스트는 호스트의 IP 주소로 구별될 수 있다. 기존의 히스토리 기반 IP 필터링에서는 이 IP 주소를 선별하는 과정에서 접근 빈도를 시간당 패킷 수와 시간당 데이터 량을 가지고 판단하였다. 앞절에서 언급한 바와 같이 이는 공격자에 의하여 위조될 소지를 안고 있다. 만일, 공격자가 앞으로 공격에 사용될 위조된 출발지 IP 주소를 정상 호스트 IP 주소에 추가할 수 있다면 보호 대상 서버는 공격에 속수무책일 것이다. 따라서, 이를 보완할 방법이 필요하며 본 논문에서는 접근 빈도외에 사용 패턴을 고려하였다.

각 서비스는 고유의 행동 패턴을 가지고 있으며 같은 서비스라고 해도 운영되고 있는 서버마다 다를 수 있다. 표 2를 보면 HTTP, SMTP, POP3, FTP 등 다양한 서비스마다 고유의 패턴을 가지고 있음을 알 수 있다. 분석 대상 서버는 TCP 프로토콜을 사용하는 서버만을 대상으로 하였다. 이 데이터는 2001년 12월 중 하루 동안 한국 IX에서 추출한 80GB의 패킷을 가지고 산출하였다. 각각의 비율은 전체 패킷 중 비율을 나타낸다.

표 1 각 서비스에 따른 SYN, ACK, FIN 비율 (단위: %)

	SYN 비율	ACK 비율	FIN 비율
HTTP	11	87	3
SMTP	37	61	6
POP3	16	83	6
FTP Control	7	97	5
FTP Data	11	89	9

예를 들어, TCP 프로토콜을 사용하는 서버를 정상적으로 이용하는 호스트들은 연결을 확립한 후 사용하게 된다. 연결 확립된 합계를 이용하여 위조되지 않은 IP 주소를 구별할 수 있다. 이와 같은 사용 패턴을 사용하여 정상 호스트 IP 주소 목록을 생성하면 공격자가 공격에 사용할 위조 주소를 고의로 주 사용자 IP에 등록하기 어려워지며 따라서 필터링의 부정적 오류가 줄어들게 된다. 정상 호스트 IP 주소 목록 선별에서는 다음의 요소들을 고려할 수 있다.

- p : 주고 받은 총 패킷 수
- s : SYN 비율
- f : FIN 비율
- t : 패킷이 조사된 일정 시간

$$c = \min(s, f)$$
$$freq = p/t$$

이 예에서는 c는 연결이 확립된 합계를 뜻하며 freq는 시간당 패킷 빈도를 뜻한다. 두 가지 값을 종합하여 일정 한계점 이상의 값을 갖는 IP 주소를 정상 호스트 IP 주소로 선정할 수 있다.

이러한 방법으로 정상 호스트 목록이 생성되었다고 가정하였을때, 생성된 정상 호스트 IP 주소 목록은 그 크기가 수 메가바이트(MB:mega bytes)에서 수십 메가바이트가 될 수 있다. 실제 사용가능한 할당된 IP 주소 대역은 약 2GB 정도이며[8], 이 중 1%(약 1000만 개)가 정상 호스트 IP 주소 목록에 포함되어 있다면 그 크기는 80MB가 되며, 0.1%가 정상 호스트라고 해도 8MB의 크기가 된다. 전 세계적으로 가장 사용자가 많은 모 사이트의 경우 하루에 접속하는 순 사용자의 비율은 전체 사용자 중 28%이다[7]. 물론 이 결과는 순위 집계에 참여한 사용자만을 대상으로 하였기 때문에 오차가 있을 수 있지만, 전 세계 인터넷 이용자 수를 5억이라 가정하면 사용자는 1.4억이며 이 사용자를 모두 목록으로 가지고 있으려면 560MB가 필요하다. 실제 DDoS 공격 발생시 공격 대상 지점에서의 대응 - 필터링 혹은 전송율 제한 - 은 한계가 있다. 따라서 주변 라우터와 분담하여 대응해야만 한다. 그러기 위해서는 정상 호스트 목록을 전달해야만 하는데 목록의 크기가 크다면 목록 전송 자체가 큰 부담이 된다. 따라서 정상 호스트 선정 기준을 더욱 엄격히 하여 목록의 크기를 줄이거나 목록을 저장하는 다른 방법을 적용하여 목록의 크기를 줄이는 것이 중요하다.

3.3 정상 호스트를 식별하기 위한 식별 집합

정상 호스트 식별 집합 생성의 목적은 공격 발생시 이 식별 집합을 비교하여 정상 트래픽을 구분하기 위함이다. 식별 집합에 따라 저장 크기와 효과가 다르게 나타난다.

정상 호스트를 식별하기 위한 집합은 크게 네 가지 형태로 정의할 수 있다. 먼저 첫 번째 형태는 정상 호스트 IP 주소를 완전히 기록하는 방식으로 목록에 포함된 정상 호스트의 수를 l이라고 할 때 $4l$ *Bytes*의 용량이 필요하다. 두 번째, 정상 호스트 IP 주소 일부를 기록하는 방법이다. 예를 들어 정상 호스트 IP 주소 하위 n비트를 기록할 수 있다. 이는 최대 $4l \cdot \frac{n}{32}$ *Bytes*의 용량이 필요하며, n 값에 따라 중복되는 호스트가 생기므로 이보다 더욱 작은 크기가 된다. 세 번째 형태는 각 정상 호스트가 보내는 패킷의 Pi 값을 기록해 두는 방법이다. 이는 총 가능한 Pi 값의 경우의 수를 k라고 했을 경우, Pi 값이 2 Bytes 이므로 최대 $4k$의 용량이 소요된다. 마지막으로 정상 호스트 IP 주소의 일부와 Pi 값을 기록하는 방법이다. 이것은 두 번째와 세 번째 방법을 병합한 방법이며 실험 결과 이 방법이 가장 효과적이며, 가장 적은 저장 용량을 차지한다. 네 번째 방법은 그림 4와 같이 표현된다. 좌측에 정상 호스트 IP 주소의 목록이 보이고, 우측에는 각 정상 호스트에 대한 Pi 값이 있다. 그리고, 상단에는 n-bit라고 표시된 부분이 있다. 식별 집합은 정상 호스트 주소 하위 n 비트와 Pi 마킹 값만을 저장하게 된다. 그림에서 진하게 표시된 부분이 저장되는 구역이다.

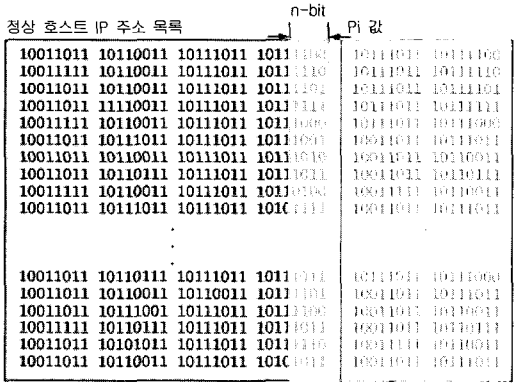


그림 4 정상 호스트 IP 주소 목록에서 정상 호스트 식별 집합 추출

각각의 네 가지 형태의 식별 집합을 사용되는 n, Pi 값에 따라 구분해서 식별 집합 (n, Pi)로 표기하며, n은 0부터 32까지의 값을 가지며, 사용되는 정상 호스트 IP 주소 하위 몇 비트를 사용하는지를 뜻한다. pi는 0 또는 1의 값을 가지며, pi가 0이면 pi를 사용하지 않겠다는 뜻이며, 1이면 pi를 사용하겠다는 뜻이 된다. 예를 들어, 식별집합 $(32, 0)$은 첫 번째 형태의 정상 호스트 IP 주소 전체를 기록하는 방법을 뜻하며, 식별 집합 $(n, 0)$은 두 번째 형태 즉, 정상 호스트 IP 주소 하위 n 비트를 기록하는 방법을 뜻한다. 식별 집합 $(0, 1)$은 정상 호스트 IP 주소를 사용하지 않고 Pi 값만을 이용하는 세 번째 형태이다. 식별 집합 $(n, 1)$은 정상 호스트 IP 주소 하위 n 비트와 Pi를 같이 쓰는 네 번째 형태를 뜻한다.

정상 호스트 식별 집합 I를 정의 하면 다음과 같다.

$$I = (l_n, l_\pi): l \in L, 0 \le n \le 32$$

L은 정상 호스트들의 집합이며, l_n은 정상 호스

트 l의 하위 n비트 값이며, l_π는 l의 Pi 값이다. n 값이 늘어날수록 정상 호스트를 정확히 식별할 수 있는 대신 저장에 필요한 용량도 따라서 늘어난다. 저장에 필요한 용량 S는 다음과 같다.

$$S = K(2 + \frac{n}{8} \cdot 2^n)\ Eytes$$

$$(where\ S \leqq 4 \mid L \mid, K = aL, 0 < a \leqq 1)$$

K는 정상 호스트들의 Pi 값 경우의 수이며, S는 $4 \mid L \mid$을 넘지 않는다.

4. 실 험

본 실험은 DDoS 공격 발생시 사용된 식별 집합, 정상 호스트의 수, 공격 호스트의 수, Pi 분포도에 따라 공격 허용율과 식별 집합의 크기를 구하기 위해 실시되었다. 식별 집합의 크기를 작게 하면서도 낮은 공격 허용율을 유지하는 것이 제안 시스템의 목표이다. 공격 허용율은 공격 호스트가 보낸 패킷 중 필터링되지 않는 패킷의 비율을 뜻하며, 정상 호스트는 이미 선정되었다는 가정하에 실험이 진행되었다.

4.1 실험 변수

실험을 위해 실제의 인터넷을 조사하여 만든 Skitter 맵[2]을 사용하였다. 이 맵에는 다수의 호스트에서 출발하여 하나의 호스트로 향하는 라우터 경로가 저장되어 있다.

표 2 실험 변수

	서버 A	서버 B	서버 C
총 호스트 수(t)	450,000	717,073	783,527
정상 호스트 수(l)	1,000~200,000	1,000~200,000	1,000~200,000
공격 호스트 수(a)	100~10,000	100~10,000	100~10,000
Pi 가지수(p)	1,802	9,872	19,917

정상 호스트 수는 1000~200,000개로 설정하고 각각에 대하여 공격 호스트의 수는 100~10,000개로 하였다. 정상 호스트와 공격 호스트는 서로 중복되지 않는다고 가정하였으며, 각 공격 호스트들은 위조된 출발지 주소를 가지고 일제히 256개의 패킷을 공격 대상으로 보낸다. Pi 마킹에 참여하지 않는 라우터의 비율은 50%로 설정하였으며 Pi 마킹에서 라우터의 식별자로는 2비트를 사용하여 Pi 값을 생성하였다.

4.2 분석

4.2.1 각 보호 대상 서버 Pi 분포도

실험에 사용된 3개의 보호 대상 서버는 서로 다른 Pi 분포도를 나타내었다. 그림 5, 6은 서버 A와 C로 임의의 호스트들이 패킷을 보냈을 경우 그 패킷이 가지는 Pi 값의 분포를 보여준다. 서버 A의 Pi 분포도는 특정 값에 편중되는 모습을 보이며, 서버 C의 Pi 분포도는 비교적 균등하게 분포되어 있다. 따라서, 서버 C의 Pi 값이 서버 A의 Pi 값보다 더 높은 분별력을 가질 것이라 예상할 수 있다. Pi 값의 분포도가 다르게 나오는 것은 각 보호 대상 서버 향하는 라우터들의 위치 구조와 라우터의 IP 주소 값들과 밀접한 관계가 있다.

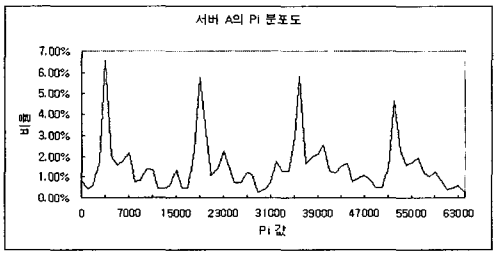


그림 5 서버 A의 Pi 분포도

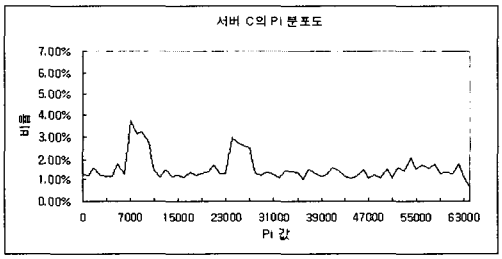


그림 6 서버 C의 Pi 분포도

4.2.2 정상 호스트의 증가에 따른 공격 허용율

서버 B에서 식별 집합 (8, 1)를 사용하여 실험한 결과, 정상 호스트의 수가 증가함에 따라 공격 허용율도 증가하였으며, 선형 비례보다는 작은 비율로 증가하였다. 식별 집합 $(n, 1)(n = 0,1,2,...,32)$의 경우에도 정도의 차이를 제외하고는 그림 7과 비슷한 향상을 보였다.

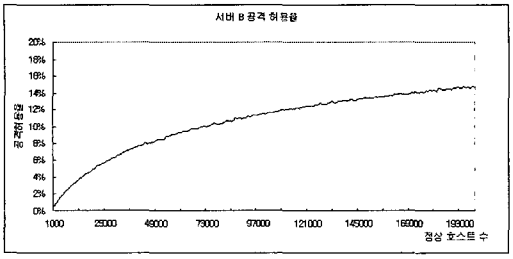


그림 7 정상 호스트 증가에 따른 공격 허용율

4.2.3 공격 호스트의 증가에 따른 공격 허용율

공격 호스트의 증가는 공격 패킷 허용율과 관계가 없었다. 그림 8는 서버 B에서 식별 집합 $(n, 1)$(n = 0, 4, 8, 12)를 사용하고, 정상 호스트의 수가 100,000일 때 공격 패킷 허용율을 보여준다. n 값에 따라 공격 허용율의 차이를 보였지만, 공격 호스트의 증가는 공격 허용율과 비교적 무관함을 알 수 있었다. 정상 호스트가 일정할 때, 공격 호스트가 허용될 확률은 확률적으로 동일하기 때문인 것으로 파악된다.

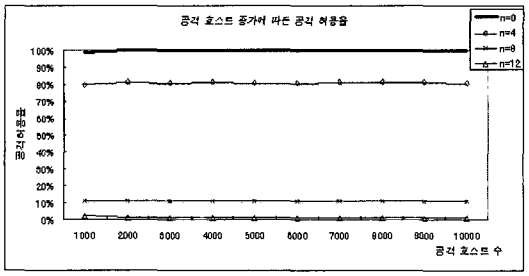


그림 8 공격 호스트 수의 증가에 따른 공격 패킷 허용 비율

4.2.4 Pi 연계시의 효과

앞서 정상 호스트 식별 집합을 정의하는 네 가지 방법중 정상 호스트 IP 주소 일부만을 기록하는 방법이 있다고 거론하였다. 실험 결과 정상 호스트 IP 주소 일부만을 가지는 것보다 Pi 값을 연계하여 가질때 더 좋은 성능을 보여주었다. 그림 9는 서버 B에 대하여 정상 호스트 200,000에 대하여 식별 집합 $(n, 1)$과 식별 집합 $(n, 0)$(n, no_pi)(n = 0,1,2,...,12)로 즉, Pi를 사용하였을 때와 그렇지 않을 때를 실험 비교하고 있다. Pi를 연계하여 사용하였을 경우 공격 허용율이 수배 이상 감소하고 있음을 보여준다. Pi의 의미는 패킷이 어느 경로로 왔음을 간접적으로 나타내며, 정상 호스트 IP 주소 하위 비트는 같은 경로에서 온 호스트를 구별하는 역할을 한다. 이로 인한 시너지 효과의 결과라고 사료된다.

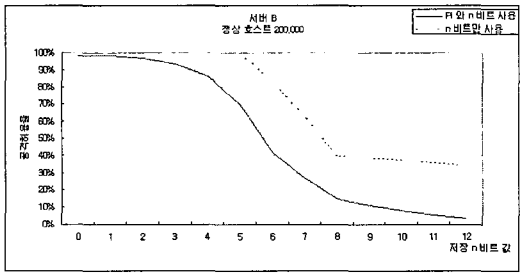


그림 9 Pi 사용시와 미 사용시 공격 허용율 비교

4.2.5 Pi 분포도에 따른 공격 허용율 비교

Pi와 정상 호스트 IP 하위 n 비트 조합으로 식별 집합을 만들고 정상호스트의 수와 n 비트 값에 따른 실험을 하였다. 두 실험 결과 모두 Pi 분포도에 따라서 성능이 확연히 차이남을 보여주었다.

그림 10은 식별 집합 (8, 1)로 실험한 결과이다. 정상 호스트의 수가 200,000일때 서버 A의 공격 허용율은 18.9%였으며 서버 C의 경우 6%였다. Pi 분포도에 따라서 3배 이상의 공격 허용율 차이가 났다.

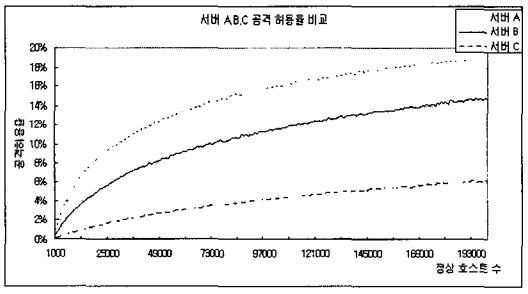


그림 10 각 서버의 정상 호스트 증가에 따른 공격 허용율

그림 11은 정상 호스트의 수가 100,000일때, 식별 집합 $(n, 1)$(n = 0,1,2,...,12)로 실험한 결과이다. 이 경우에도 서버 C가 가장 좋은 성능을 보여주었으며 Pi분포도에 따른 결과라고 판단되었다. Pi 값이 고르게 분포되어 있을 경우 공격 허용율이 낮았으며, 이는 Pi 값을 고르게 분포하기 위하여 개선된 Pi 방법이 필요함을 보여주었다.

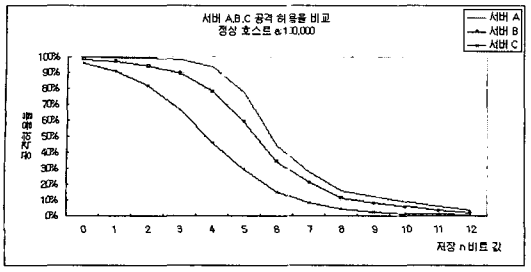


그림 11 각 서버의 공격 허용율 비교

4.2.6 n 값에 따른 공격 허용율과 저장 크기

지금까지의 실험 결과 정상 호스트의 수가 늘어날수록 공격 허용율이 상승하였다. 선형비례보다 증가폭이 적은 형태가 나타났다. 또한, 식별 집합 $(n, 1)$에서 사용된 n 값이 커지고 Pi 분포도가 좋을수록 공격 허용율이 낮았으며, n 값이 높아질 수록 저장 크기 또한 증가하였다. 따라서, 저장 크기와 공격 허용율을 비교하여 적정 수준의 r. 값을 취하는 것이 필요하다. 그림 12는 서버 A의 정상 호스트가 100,000(a)일 경우와 200,000(b)일 경우의 저장 용량과 공격 허용율의 관계를 보여주고 있다. 저장 크기는 식별 집합 (32, 0)을 사용한 저장 크기 대비 비율을 나타낸다. 공격 허용율이 급감하며 저장 크기도 일정 수준을 유지하는 식별 집합 (8, 1)를 선택하는 것이 타당하다고 판단할 수 있다. 이와 같이 보호하려는 대상 서버의 정상 호스트 수, Pi 분포도를 고려하여 적절한 식별 집합을 선택해야 한다.

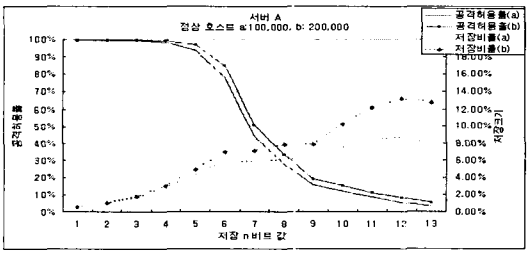


그림 12 정상 호스트 IP 주소 하위 n 비트 값 사용에 따른 공격 허용율과 저장 크기

5. 결 론

인터넷 인프라가 비약적으로 발전하고 있지만 관련 보안 기술은 이를 따라가지 못하고 있는 실정이다. 특히, 인터넷의 전달 매체인 패킷에 대한 인증, 서명, 부인 방지, 권한 제어 등의 기능이 없기 때문에 패킷을 보낸 주체를 확신할 수 없으며 공격자를 비롯한 모든 호스트는 제한 없이 패킷을 원하는 어떤 곳에도 보낼 수 있다. 이러한 취약성과 인터넷의 비약적 발전이 서로 맞물려 DDoS 공격의 위험성을 더욱 높여주고 있으며, 새로운 DDoS 공격이 계속적으로 등장하고 있다.

본 논문에서는 DDoS 공격 발생시 이에 즉각적으로 대응하며 새로운 공격 형태를 가지는 DDoS 공격에 적응력 높은 시스템을 제안하였다. 보호 대상 서버를 이용하는 정상 호스트들의 트래픽을 보호하기 위한 방법에 초점을 맞추어, 사용 패턴을 추가하여 개선시킨 히스토리 기반 IP 필터링을 제시하고, 이로부터 얻어진 정상 호스트 목록을 효과적으로 이용하기 위하여 Pi를 도입하였다. 제안된 시스템은 결과적으로 적은 용량의 필터링 정보 - 정상 호스트 식별 정보 - 를 가지고 낮은 공격 허용율을 보여주었다.

본론에서도 언급하였듯이 보호 대상 서버의 환경에 따라 Pi 값이 특정 값에 편중되는 현상을 보였다. 이는 대상 서버를 보호하기에 악조건이 되므로 Pi 값을 균등하게 분포시킬 수 있는 방법을 연구하는 것이 필요하다. 또한, 한 무리의 경계 라우터들이 하나 이상의 보호 대상 서버와 협업할 경우 생길 수 있는 문제를 해결하여 범용적으로 사용될 수 있는 시스템을 개발할 수 있다.

참고 문헌

[1] Abraham Yaar, Adrian Perrig, Dawn Song, "Pi: A Path Identification Mechanism to Defend against DDoS Attacks," Security and Privacy, 2003. Proceedings. 2003 Symposium on, May, 11-14, 2003 Page(s): 93-107

[2] Caida. Skitter. http://www.caida.org/tools/measurement/skitter/, 2000

[3] Chang, R.K.C., "Defending against flooding-based distributed denial-of-service attacks: a tutorial," IEEE Communications Magazine, Volume: 40 Issue: 10 , Oct. 2002 Page(s): 42-51

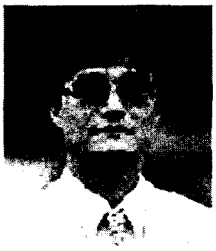
[4] Christoph L. Schuba, Ivan V.Krsul, Markus G.Kuhn, Eugene H.Spafford, Aurobindo Sundaram, Diego Zamboni, "Analysis of a

Denial of Service Attack on TCP," 1997 IEEE
[5] "Denial of Service Attacks," CERT, 1997
[6] http://staff.washington.edu/dittrich/misc/ddos/, "Analyses and talks on attack tools" 1999 2000
[7] http://www.alexa.com, 2003
[8] Internet Assigned Numbers Authority, http://www.iana.org
[9] Jaeyeon Jung, Balachander Krishnamurth, and Michael Rabinovich. "Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites." WWW10 WWW2002, May, 7-11, Honolulu, Hawaii, USA 2002.
[10] Jelena Mirkovic, Gregory Prier, Peter Reiher, "Attacking DDoS at the source," Proceedings of the 10 th IEEE International Conference on Network Protocols (ICNP'02) 2002 IEEE
[11] Jelena Mirkovic, Peter Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," ACM, p.67, 2002
[12] John Ioannidis, Steven M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," Network and Distributed System Security Symposium, February, 2002.
[13] Jun Xu; Wooyong Lee, "Sustaining availability of web services under distributed denial of service attacks," Computers, IEEE Transacti on, Volume: 52 Issue: 2 , Feb. 2003 Page(s): 195-2
[14] Kashiwa, D. Chen, E.Y.; Fuji, H, "Active shaping: a countermeasure against DDoS attacks," Universal Multiservice Networks, 2002. ECUMN 2002. 2nd European Conference on, 2002 Page(s): 171 -179
[15] Kevin J. Houle, George M. Weaver, "Trends in Denial of Service Attack Technology," CERT Coordination Center, October, 2001
[16] K. Park and H. Lee, "A Proactive Approach to Distributed DoS Attack Prevention using Route-Based Distributed Filtering," Tech. Rep. CSD-00-017, Department of Computer Sciences, Purdue University, December, 2000
[17] Kihong Park, Heejo Lee, "On the Effectiveness of RouteBased Packet Filtering for Distributed DoS Attack Prevention in PowerLaw Internets," SIGCOMM'01, August, 2731, 2001, San Diego, California, USA.
[19] K. L. Calvert, "Active Networking Working Group," University of Kentucky, RFC Draft, July, 1999
[20] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker, "Controlling High Bandwidth Aggregates in the Network."
[21] Ryan Naraine, "Massive DDoS Attack Hit DNS Root Servers," eSecurity Planet.com 2002. 10. http://www.esecurityplanet.com/trends/article.php/10751_1486981.
[22] Sourcefire. "Snort: The Open Source Network Intrusion Detection System."
[23] Stefan Savage, David Wetherall, Anna Karlin, Tom Anderson, "Practical Network Support for IP Traceback," SIGCOMM'00, ACM, 2000
[24] Shalaby, N.; Peterson, L.; Bavier, A.; Gottlieb, Y.; Karlin, S.; Nakao, A.; Xiaohu Qie; Spalink, T.; Wawrzoniak, M., "Extensible routers for active networks," Page(s): 92- 116
[25] Sterne, D.; Djahandari, K.; Balupari, R.; La Cholter, W.; Babson, B.; Wilson, B.; Narasimhan, P.; Purtell, A.; Schnackenberg, D.; Linden, S., "Active network based DDoS defense," Page(s): 193- 203
[26] Tao Peng, Leckie, C., Ramamohanarao, K. "Protection from distributed denial of service attacks using history-based ip filtering," Communications, 2003. ICC '03. IEEE International Conference on, Volume: 1 Page(s): 482-486, 2003

정 형 철

2002 아주대학교 정보통신공학부(학사)
현재~아주대학교 정보통신전문대학원 석사과정
관심분야 : 인터넷 면역 시스템, 임베디드 시스템
E-mail positif@ajou.ac.kr

홍 만 표



1981 서울대학교 계산통계학과(이학사)
1983 서울대학교 계산통계학과(이학석사)
1991 서울대학교 계산통계학과(이학박사)
1983~1985 울산공과대학 전자계산학과 전임강사
1985~현재 아주대학교 정보 및 컴퓨터 공학부 교수
1993~1994 미네소타대학 전자공학과 교환 교수
관심분야 : 병렬처리
E-mail : mphong@ajou.ac.kr