DDoS 공격 탐지와 대응에 관한 연구 : FDDS(Flow-based DDoS Detection System)

권윤주*, 문정훈*, 이만희*, 변옥환*

*한국과학기술정보연구원, 슈퍼컴퓨팅인프라개발실

Study on Detection and Reaction of DDoS Attack

Yoon Joo Kwon*, Jeung Hoon Moon*, Manhee Lee*, Ok Hwan Byeon

*Department of Supercomputing Infra Development, Korea Institute of Science Technology Information

요 약

최근 인터넷을 통한 해킹이나 바이러스 침투로 인한 사례들이 증가하고 있다. 2000년 2월, Yahoo, Amazon, CNN에 발생했던 DDoS 공격으로 인해 각 웹 사이트들은 큰 피해를 입었다. 인터넷의 개방성은 사용자들에게 매우 다양한 서비스를 제공하는 반면, 인터넷을 통한 해킹, 바이러스 등의 공격을 위한 도구로서 사용되고 있다. 본 논문은 근래 분산서비스거부 (DDoS) 공격으로 인하여 남용되고 있는 네트워크 자원의 손실을 감소시키기 위해서, 분산서비스거부 (DDoS) 공격을 탐지하고 그 공격에 대해 적절한 대응조치를 취할 수 있는 시스템인 FDDS (Flow based DDos Detection System)를 제안한다.

I. 서론

최근 인터넷을 통한 해킹이나 바이러스 침투로 인한 사례들이 증가하고 있다. 2000년 2월, Yahoo, Amazon, CNN에 발생했던 분산서비스거부 (DDoS : Distributed Denial of Service) 공격으로 인해 각 웹 사이트들은 큰 피해를 입었다. 이러한 서비스거부 (DoS : Denial of Service) 및 분산서비스거부 (DDoS : Distributed Denial of Service) 공격의 방법과 유형은 날로 다양해지고 있으며, 대용량 트래픽 공격을 통해 전세계 네트워크가 중단되는 사례도 급속히 증가하고 있다.

이렇듯 인터넷의 개방성은 사용자들에게 매우 다양한 서비스를 제공하는 반면, 인터넷을 통한 해킹, 바이러스등의 공격을 위한 도구로서 사용되고 있다. 해커의 공격을 통해 시스템을 마비시키거나 시스템에 저장되어 있는 데이터 안정성과 무결성을 침해하는 보안 위험은 대기업, 서비스 제공 업체들이 현재 직면하고 있는 가장 중요한 문제이며 이를 해결하기 위한 방법으로 방화벽 및 IDS의 도입이 늘어나고 있지만 서비스거부공격(DoS) 및 분산서비스거부 (DDoS) 공격은 이러한 격리된 솔루션으로 해결할 수 있는 규모의 공격이 아니다.

본 논문은 시스템 파괴 및 네트워크 자원 고갈을 유도하는 해킹을 저지할 수 있는 네트워크 보안 방법으로 FDDS (Flow-based DDoS Detection System)를 제시하고자 한다. 서비스거부 (DoS) 공격을 포함한 분산서비스거부 (DDoS) 공격은 일반 해킹과는 달리 시스템 파괴의 목적뿐만 아니라 네트워크 자원 점유를 통한 서비스 차단에 목적이 있기 때문에 FDDS는 중간단계인 백본에서의 비정상 트래픽 차단을 통하여 분산서비스거부 (DDoS) 공격에 대한 적절한 대응책을 제공하고자 한다.

본 논문은 다음과 같이 구성되어있다. 본론에서는 관련연구와 제안하고자 하는 시스템인 FDDS에 대하여 기술할 것이며, 마지막으로 결론과 향후 계획에 대하여 서술할 것이다.

II. 본론

1. 관련연구

1) DDoS(Distributed Denial of Service)

DDoS 공격은 인터넷에 연결된 일단의 시스템들을 이용해 단일 사이트에 대한 Flood 공격을 시도하는 것이다[4]. 해커들이 일단 취약한 인터넷 시스템에 대한 액세스가 이뤄지면 침입한 시스템에 소프트웨어를 설치하고 이를 실행시켜 원격에서 공격을 개시한다. DDoS 공격을 개시하는데 사용되는 프로그램으로는 TrinOO, TFN(Tribe

Flood Network), TFN2K, Stacheldraht 등이 있다.

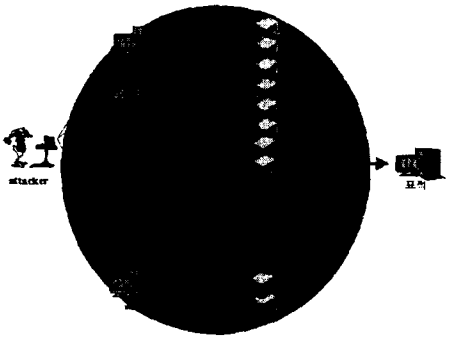

그림 1. DDoS 공격의 구조

DDoS 공격은 그림 1과 같은 구조로 하나의 시스템을 공격한다. 그림 1에서 보는 바와 같이 DDoS 공격은 Attacker, Master, Agent로 구성된다. 여기서 Agent들은 DDoS 공격을 위한 침해호스트로서, Attacker들은 수백에서 수천대의 Agent를 필요로 한다. 하나의 호스트를 침해시키는 과정이나 도구를 설치하는 일들은 자동화되어 있으며 다음과 같은 단계로 나누어져 있다[6].

① 알려진 취약성 조사를 위한 수많은 호스트에 대한 스캔 시도

② 접근 권한 획득을 위한 취약한 호스트 공격

③ 대상 침해시스템에 도구 설치

④ 추가적인 스캐닝 및 침해 공격을 위해 침해시스템 이용

Master는 침해된 호스트들인 Agent를 관리하면서 Attacker로부터의 명령이 오면 자신이 관리하고 있는 Agent들을 이용하여 하나의 target system을 공격한다.

target system을 공격하는 방법은 다양하며, 다음의 DDoS 공격의 유형에서 설명하기로 한다.

2) DDoS(Distributed Denial of Service) 공격의 유형

DDoS 공격의 유형은 표1과 표2에서 보는 바와 같이 대표적으로 6가지 정도 발견되었다. 여기서는 Trinoo, TFN, Stacheldraht에 대해서 설명하기로 한다.

종류 / 항목	trinoo	TFN	Stacheldraht
공격방법	UDP floods	UDP/SYN/ICMP floods, Smurf	UDP/SYN/ICMP floods, Smurf
통신암호화기능	X	X	O
Attacker <-> Master	27665/tcp	TELNET등의 방법 (별도의 연결 없음)	16660/tcp (암호화)
Master <-> Attacker	27444/udp	ICMP echo reply	ICMP echo reply 65000/tcp(unused)
Agent <-> Master	31335/udp	ICMP echo reply	ICMP echo reply
IP spoofing 기능	X	O	O
발견시기	1999 이전	1999 이전	1999/08

표 1. DDoS 공격의 종류(I)

• Trinoo

Trinoo는 1999년 7월에 처음으로 발견되었다. 공격 방법으로는 UDP flood를 사용하고, IP Spoofing은 사용하지 않는다[1][5].

• TFN

"Tribe Flood Network"의 약자로서, Mixer라는 독일의 해커에 의해서 개발되었다. TFN의 Trinoo와는 달리 Attacker가 Master로 접속하기 위한 별도의 Port 번호가 준비되어 있지 않다. 따라서 공격자는 Master로 접근하기 위해 TELNET등의 프로그램을 사용해서 Master를 구동시켜야 한다. 공격방법은 매우 다양하며, ICMP/TCP SYN/UDP floods, 그리고 Smurf 공격이 가능하다.

Master와 Agent의 통신에 ICMP ECHO_REPLY 메시지를 사용하므로 별도의 Port 번호를 열어둘 필요가 없어서 쉽게 탐지되지 않는다[1][5].

• Stacheldraht(barbed wire : 철조망)

Sstacheldraht는 "Trinoo"의 네트워크 구조와 "TFN"의 다양한 공격방법(TCP SYN/UDP/ICMP floods, Smurf 공격) 그리고, 통신상의 암호화기능을 포함한 DDoS 공격도구이다[1][5].

암호화를 위해서 Attacker가 직접 사용하는 TELNET과 비슷한 프로그램을 제공하는 데 이 프로그램이 공격자와 Master간의 암호화된 통신을 보장한다. 다른 공격들과 특히 다른 점은 IP Spoofing 기능을 위해서 사전에 테스트를 실시한다는 것이다[5].

종류 / 항목	Shaft	Mstream	TFN2K
공격방법	UDP/SYN/ICMP floods, Smurf	TCP ACK floods	UDP/SYN/ICMP floods, Smurf
통신암호화기능	X	X	O
Attacker <-> Master	20432/tcp	6723/tcp (변종 : 15104/tcp)	TELNET등의 방법 (별도의 연결 없음)
Master <-> Attacker	18753/udp	7983/udp (변종 : 10498/ udp)	UDP/TCP/ICMP 를 random하게 사용(암호화)
Agent <-> Master	20433/udp	9025/udp (변종 : 6838/udp)	없음
IP spoofing 기능	X	O	O
발견시기	1999/11	2000/03	1999/11

표 2. DDoS 공격의 종류(II)

3) DDoS를 방어하는 데 있어서 문제점

• DDoS Stream에는 필터링하거나 탐지할 수 있는 일반적인 특성이 없다[1][2]. 결국 DDoS 공격으로 전송되는 네트워크 트래픽은 서비스의 합법적인 사용을 위한 트래픽과 구분되지 않는다[4].

• 분산된 DDoS 공격 근원지간의 협동은 DDoS 공격을 trace back 하기 어렵게 한다. 따라서 분산된 근원지에 대처할 수 있도록 관리 도메인간 협력이 필요한 반면 관리 도메인 간의 협력이 부족하다[1][2].

• DDoS 공격 코드와 자동화된 툴은 인터넷으로부터 쉽게 다운받을 수 있어서 초보 해커(intruder)도 쉽게 강력한 공격을 실행시킬 수 있다[1][2].

• Attacker는 attacking machine의 신분(identity)를 숨기기 위해 IP Spoofing을 사용하기 때문에 공격하는 시스템의 신분(identity)을 알아내기 어렵다[1][2][4].

• 인터넷 호스트에는 꾸준히 Security Hole이 존재한다[1][2].

2. FDDS (Flow-based DDoS Detection System)

DDoS 공격은 전술한 바와 같이 그 특성이 명확히 존재하지 않는다. 따라서 DDoS 공격은 주기적인 네트워크 모니터링을 통하여 비정상적인 네트워크 트래픽을 탐지해 내어야 한다.

일반적으로 네트워크 트래픽은 Service 요청/응답으로 이루어지기 때문에 inbound 트래픽과 outbound 트래픽의 비율은 비슷하다. 그러나 플러드형 공격이 시작되면 그러한 트래픽간 균형이 깨져서 어느 한 쪽으로 트래픽이 몰리는 경향을 볼 수가 있다.

본 논문은 이러한 점에 착안하여 네트워크 트래픽을 연속적으로 관찰하면서 그 정상적인 트래픽의 behavior를 알아내고 비정상적인 트래픽을 구분해 낼 수 있는 시스템을 구현하였다.

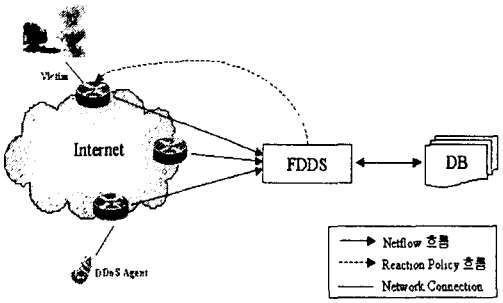

그림 2. FDDS의 구성

그림 2는 본 논문에서 제기하고자 하는 FDDS의 전반적인 구성을 나타낸다. DDoS 공격의 경우 대단위 트래픽량으로 하나의 네트워크 또는 시스템 자원을 고갈시키는 방법을 취하기 때문에 이러한 DDoS 공격을 탐지하기 위해서는 네트워크 전체에 대한 거시적인 트래픽 모니터링이 필요하다. 일반적으로 단위 네트워크마다 하나 이상의 라우터가 그 네트워크에 대한 트래픽 조율을 담당하기 때문에, 그림 2에서 보는 바와 같이, FDDS는 각 네트워크 트래픽을 조율하는 라우터로부터 하나의 네트워크로 유입 또는 유출되는 트래픽의 정보를 받는다. 이때 라우터로부터 전송되어지는 트래픽 정보가 'Netflow'이다. FDDS는 라우터들로부터 Netflow 정보를 받아, 이를 가공하여 네트워크 트래픽의 behavior를 관찰하고, 네트워크의 비정상 트래픽을 탐지하면 해당 라우터에 Reaction Policy를 보내 제어한다.

이러한 FDDS의 세부 요소들은 그림 3과 같이, Netflow Collector, Netflow Classfier, Netflow Analyzer, Traffic Checker, Reactor로 구성되어 있다.

Netflow Collector는 해당 라우터로부터 주기적으로 Netflow를 exporting 받아서 라우터로부터의 Netflow 데이터를 FDDS에서 분석할 수 있도록 플로우별로 각 세부 요소들을 분리하여 수집한다. Netflow의 세부 요소로는 근원지 주소, 목적지 주소, 프로토콜 번호, 서비스 포트 번호, 라우터 주소, 플로우당 packet량, 플로우당 byte량 등이 있다.

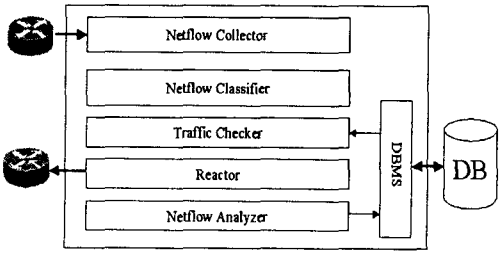

그림 3. FDDS의 세부 요소

Netflow Classifier는 Netflow Collector를 통해 수집된 네트워크 트래픽 자료를 각 플로우의

inner interface, outer interface의 조합을 통해서 각 플로우에 대한 inbound와 outbound 트래픽을 계산한다. inbound와 outbound로 나누어진 각 플로우 자료들은 protocol 번호에 의해서 inbound_tcp, inbound_udp, inbound_icmp, inbound_else, outbound_tcp, outbound_udp, outbound_icmp, outbound_else로 분류되어 각각의 통계치를 구한다.

Traffic Checker는 기존의 평균값과 표준편차를 통하여 신뢰구간 95%로 트래픽의 normal과 abnormal 트래픽을 구분한다. Traffic Checker에서는 현 트래픽이 abnormal 트래픽으로 판정되면, 분산서비스거부 (DDoS) 공격여부를 재차 검사하기 위해서 해당 분류단위별로 근원지 주소/목적지 주소의 top10을 구한다. 플로우 개수를 기준으로 정렬된 목적지 주소별 1위가 전체 트래픽의 90%이상을 분산서비스거부 (DDoS) 공격으로 간주한다.

Reactor는 Traffic Checker에서 현 트래픽에 대해서 분산서비스거부 (DDoS) 공격 판정이 나왔을 때 해당 라우터로 제어 정책을 내리는 모듈이다. 이때Reactor모듈은 Traffic Checker에서 플로우 개수를 기준으로 정렬한 목적지주소를 이용하여 라우터에 AccessList를 작성한다.

Netflow Analyzer는 전체 트래픽에 대한 정상 트래픽 모델 및 각 프로토콜별 트래픽 모델을 수정하기 위해서 기존 평균 및 표준편차값에 현 플로우의 값을 포함하여 다시 계산한다.

Reactor에서는 그림 4 에서 보는 바와 같이, 라우터에 ratelimit를 적용시키기 위한 Network Traffic State를 관리한다. Network Traffic State는 다음의 3가지 상태를 가지고 있다.

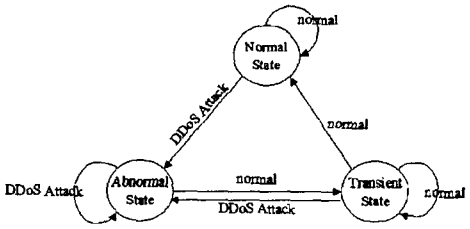

그림 4. 네트워크 트래픽의 State Diagram

- Normal State
- Abnormal State
- Transient State

여기서 Normal State는 트래픽이 정상적인 흐름을 가지고 있는 상태이며, Abnormal State는 트래픽이 비정상적인 흐름을 가지고 있는 상태이다. Transient State는 Abnormal State에서 Normal State로 가기 전단계로서 네트워크 트래픽 상태가 안정되고 있는 지 한 번 점검하는 단계이다.

Normal State에서 현 트래픽이 Traffic Checker에 의해 분산서비스거부 (DDoS)공격 판정을 받으면 Reactor는 Network Traffic State를 Normal State에서 Abnormal State로 이동시킨다. Abnormal State는 비정상ㅇ적인 네트워크 트래픽 흐름을 의미하는 상태로서, Abnormal State에서 Reactor의 Action은 라우터에 ratelimit을 적용시키는 것이다. Reactor에서하는 ratelimit는 Traffic Checker에서 구해진 90%이상의 트래픽을 가진 목적지 주소에 적용된다.

Abnormal State에서 들어오는 트래픽이 Traffic Checker에 의해 연속적으로 분산서비스거부 (DDoS) 공격 판정을 받으면 Reactor는 Network Traffic State를 Abnormal State에서 Normal State로 천이시킬 수 없다. Abnormal State에서 현 트래픽이 Traffic Checker에 의해 정상판정을 받으면 Reactor는 Normal State로 상태천이를 해도 되는지를 재차 검사하기 위해서 Transient State로 Network Traffic State를 이동시킨다. 이때에도 ratelimit는 계속 적용된다. 이후 다시 정상 판정을 받게 되면 Network Traffic State를 Normal State로 이동시키고 ratelimit 적용을 폐기한다.

III. 결론 및 향후 계획

근래 대두되고 있는 분산서비스거부 공격은 단지 시스템 파괴의 목적을 지니고 있는 것이 아니라 네트워크 자원을 잠식하여 네트워크 서비스 자체가 되지 못하도록 한다. 이러한 분산서비스거부 공격은 따라서 네트워크 보안측면에서 뿐만이 아니라 Congestion Control 면에서도 이 문제를 풀어가려는 노력들이 이루어지고 있다.

이러한 공격에 대비하기 위한 시스템으로 본 논문에서는 FDDS(Flow-based DDoS Detection Service)를 제안했다. FDDS는 네트워크 상의 트래픽을 신속하고 정확하게 수집하기 위해서 라우터에서 생성되는 Netflow 데이터를 이용한다. Netflow 데이터는 라우터를 지나는 트래픽을 플로우 단위로 정리한 자료로서 DDoS 공격의 경우 패킷량의 증가뿐만이 아니라 플로우량의 급속한 증가 경향을 탐지해 낼 수 있는 데이터로 사용된다. FDDS는 네트워크 트래픽의 상태를 Normal State, Abnormal State, Transient State로 분류하여 관리하여 네트워크 트래픽 상태에 맞게 대처함으로써, 네트워크를 분산서비스거부(DDoS)에 대하여 안정적으로 관리할 수 있다.

본 논문에서는 FDDS를 이용하여 DDoS 공격을 탐지하고 대응하는 것에 대한 전체적인 구조에 대하여 논하였는데, 좀 더 많은 테스트를 통하여 비정상트래픽에 대한 모델링이 앞으로 필요할 것으로 사료된다. 따라서 앞으로 FDDS를 더욱 세부적으로 개발하여 나갈 것이며, 네트워크 트래

픽에 대한 모델링에 대하여 연구할 것이다.

참고문헌

[1] Jelena Mirkovic, "D-WARD : DDoS Network Attack Recognition and Defense", PhD Proposal, 2002년 1월.

[2] Jelena Mirkovic, "Source Router Approach to DDoS Defence", Usenix Security Symposium 2001, 2001

[3] Dan Sterne, et al., "Active Network Based DDoS Defense", DANCE02(DARPA Active Network Conference and Exposition), pp. 193-203, 2002

[4] Felix Lau, et al., "Distributed Denial of Service Attacks", Systems, Man, and Cybernetics, 2000 IEEE International Conference on, Vol. 3, pp 2275-2280, 2000

[5] 이철호, "DDoS 공격도구 분석", 2002년 2월 , http://rpa.ajou.ac.kr/project/linux_q_team/seminar_doc/L_ts020204-DDOS공격도구분석.pdf

[6] White Paper "Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks", CISCO, 2000년 2월