DDoS 공격에 대한 탐지 및 추적시스템 제안

이근수*, 박지현*, 장진용*, 송주석*, 유동영**

*연세대학교, **한국정보보호진흥원

A Proposal for Detection and Traceback System against DDoS Attack

Keun-Soo Lee*, Ji-Hyun Park*, Jin-Yong Jang*, Joo-Souk Song*, Dong-Young Yoo**

*Dept. of Computer Science, Yonsei Univ., **Korea Information Security Agency

요 약

인터넷 기술과 시장의 급성장으로 인터넷통신은 우리 생활속에 크게 자리잡고 있다. 그런 만큼 인터넷으로부터 얻는 정보는 가히 무시할 수 없을 정도이다. 이런 환경에서 최근 웹사이트의 정상적인 서비스를 방해하는 DDoS 공격은 공격방법과 은닉기법이 날로 첨단화·다양화되고 있어, 이에 대한 탐지와 근원지 추적이 어려운 상태이다.

따라서 본 논문에서는 DDoS 공격의 중간 계층인 핸들러와 에이전트를 추적하기 위해, 핸들러·에이전트간의 통신 취약점을 이용, 이를 탐지하고 역추적할 수 있는 시스템을 제안하고자 한다.

I. 서론

얼마 전, 분산 서비스 거부(DDoS)라고 불리는 일련의 공격들로 인하여 유명 웹사이트들은 많은 피해를 당했다. 이 공격은 대상 사이트에 불필요한 트래픽을 과도하게 발생시켜 정상적인 사용자의 서비스를 막으며, 일반적인 DoS와는 달리 다수의 공격지점을 갖기 때문에, 피해가 심각하고 공격지점을 찾기도 어렵다. 현재, DDoS 공격을 자동으로 수행해주는 여러 가지 도구들이 나와 있으며, 이에 대한 분석을 통해 DDoS 공격의 특징 및 약점들이 드러나고 있다. DDoS 공격에 대한 대책으로는 공격 패킷을 추적하는 기법, 공격 패킷을 막는 기법, 공격 도구를 제거하는 기법 등 많은 방법들이 연구되고 있으며, 각 기법들은 상호보완적으로 동작할 수 있다.

DDoS의 계층 구조에서 중요한 역할을 차지하는 핸들러와 에이전트는 상호 동작하는 구조를 갖고 있다. 따라서 그들간의 통신 특징 및 취약점을 파악한다면, 이를 통해 각각을 찾아내고 역추적할 수 있다. 본 논문에서는 이러한 핸들러·에이전트간의 통신에서 나타나는 취약점을 분석하고, 역추적을 할 수 있는 DDoS 탐지 및 추적 시스템을 제안하는 것을 목표로 한다. 논문의 구성은 다음과 같다. 2장에서는 DDoS의 기본 구조 및 각각의 특징적인 통신특성을 분석한다. 3장에서는 분석된 내용을 바탕으로, DDoS 탐지 및 추적시스템을 제안한다. 4장에서는 이에 대한 결론 및 향후 연구로 끝을 맺는다.

II. DDoS 분석

1. DDoS의 계층 구조

DDoS의 계층간 통신특성을 분석하기에 앞서, DDoS 공격의 특징적인 구조를 살펴본다. 일반적으로 DDoS 공격은 다계층 구조를 갖는 공격 모델을 갖는다. 기본 공격 모델을 이루고 있는 4가지 구성요소는 다음과 같다[1].

o 공격자 : 실제적인 공격을 지휘하는 은닉자

o 핸들러 : 공격자가 공격을 명령할 때 사용하

본 논문은 한국정보보호진흥원 위탁과제로 수행 중인 "DDoS 공격자 근원지 분석기술 연구 및 프로토타입 구현" 결과의 일부임.

는 소프트웨어(마스터)가 설치된 시스템

o 에이전트 : 실제적인 공격패킷을 전송하는 소프트웨어(데몬)가 설치된 시스템

o 희생자(Victim) : 공격을 당하는 대상시스템

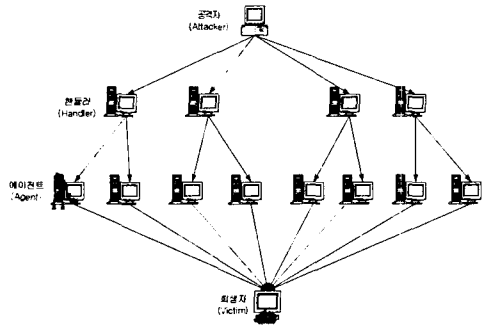

그림1 : DDoS의 공격 계층 구조도

그림 1에서 보면, 공격자는 핸들러에 DDoS 공격 도구를 설치하고, 핸들러는 에이전트에게 공격을 명령한다. 공격 명령을 받은 에이전트들은 동시에 패킷을 플러딩함으로써, 희생자(victim) 시스템을 마비시킨다.

에이전트의 개수는 수 천 개까지도 가능하다. 하나의 에이전트는 분당 수 천 개의 패킷을 대상 시스템 전송할 수 있다. 100개의 에이전트가 있을 경우는, 분당 수 백만 개의 패킷이 전송되어 대상 시스템이 갖고 있는 모든 가용한 대역폭을 소비시킬 수 있다. 이 때, 핸들러와 에이전트까지의 경로는 DDoS 공격 도구의 다양한 특징이 나타나는 부분이다. 즉, 공격자는 다양한 DDoS 공격 도구 중에서 하나를 선택하여 핸들러에 설치하게 되며, 각각의 공격 도구들은 핸들러와 에이전트간 통신에 있어서 고유의 통신 특성을 가지게 된다.

2. DDoS의 통신특성

여기서는 현재 알려진 몇몇 DDoS 공격도구의 마스터 · 데몬간 통신에서 나타나는 특성을 분석한다.

1) Trinoo

trinoo에서 마스터 · 데몬간 통신은 udp 포트를 사용해 이루어진다. 마스터 프로그램을 동작시키거나, 데몬에게 명령을 보내기 위하여 암호를 필요로 한다. 이런 암호들은 별도의 조작없이 평문 형태로 보내져, 프로그램 내부에 저장된 crypt() 형태의 암호문과 비교된다. 알려진 통신으로는 데몬이 시작될 때 *HELLO* 메시지를 마스터에게 보내며, 마스터의 png에 대해서 데몬은 PONG의 응답을 보낸다[2].

2) TFN(Tribe Flood Network)

TFN에서 마스터 프로그램은 명령어 라인으로 동작하며, 실행하기 위한 암호는 필요하지 않다. 공격명령을 수행하게 될 데몬들의 리스트는 'iplist' 라는 별도의 파일로 존재한다. TFN에서의 마스터 · 데몬간 통신은 TCP나 UDP가 아닌 ICMP ECHOREPLY 패킷을 통하여 이루어진다. 데몬에게 보내지는 명령은 ICMP 패킷의 id 필드에 16비트 이진형태로 암호화되며, 이는 설정파일에서 변경할 수 있다. 데몬을 통한 공격에서 소스 IP와 소스 포트는 임의로 주어지며, 패킷의 크기도 변경이 가능하다[3].

3) Stacheldraht

Stacheldraht에서의 마스터 · 데몬간 통신은 포트번호 65000의 TCP와 ICMP를 사용한다. TCP는 데이터 통신에 사용되며 Blowfish로 암호화된다. ICMP는 몇몇 기능들을 위해 사용되며 payload는 암호화되지 않는다. 데몬이 시작될 때, 마스터들의 IP 주소를 포함하는 리스트파일을 검색하며 이 파일은 Blowfish로 암호화되어 있다. 파일을 찾지 못하는 경우에는 데몬 내부적으로 컴파일된 마스터의 IP주소를 사용한다. 마스터와 데몬은 ICMP ECHOREPLY 패킷을 주기적으로 주고받는 특징을 갖는다[4].

III. 탐지 · 추적 시스템 제안

1. 통신취약점을 활용한 탐지 및 추적

2장에서 분석한 결과를 통해, 각 DDoS 공격도구의 통신에서 나타난 취약점을 활용한 탐지기법과 추적방안은 다음과 같다.

1) Trinoo

trinoo는 기본적으로 큰 수의 udp 포트번호를 사용하며, 이에 대한 모니터링이 필요하다. 이미 알려진 png, PONG, *HELLO* 등의 문자열을 udp 패킷에서 찾는 방법을 쓸 수 있다. 마스터를 발견한다면 데몬의 리스트가 들어있는 파일을 찾거나, bcast 명령을 사용하여 데몬의 리스트를 얻을 수 있다. 데몬을 발견한다면 'strings' 명령어를 통해 마스터의 IP 리스트를 얻을 수 있다.

2) TFN

네트워크에 대한 모니터링을 통해 공격에 사용된 ICMP 패킷을 찾아내는 것이 중요하다. 데몬에서 마스터로 보내지는 ICMP 패킷의 payload에는

공격명령에 대한 결과가 직접 나타나기 때문에 이를 검사하여 공격에 사용된 패킷인지를 확인할 수 있다. 또한, 분석된 TFN 소스코드에서 ICMP ECHOREPLY 패킷헤더에 사용하는 시퀀스 번호는 항상 0인 것도 이용할 수 있다. 마스터를 발견한다면 iplist 파일을 통해 데몬들을 찾을 수 있다. 데몬을 발견한다면 데몬이 보내는 ICMP 패킷을 통해 마스터를 찾을 수 있다.

3) Stacheldraht

TCP를 통한 통신은 암호화되어 있기 때문에, trinoo에서처럼 ICMP 패킷에 대한 모니터링을 이용한다. stacheldrat 데몬은 IP 스푸핑이 가능한지를 알아보기 위하여 ICMP 패킷을 마스터에게 보내게 되는데, ID필드 번호 및 데이터 필드에 포함되어 있는 'skillz', 'spoofworks' 등을 확인함으로써 공격패킷을 알아낼 수 있다. 마스터를 발견한다면 showalive, showdead 명령어를 통해 데몬들의 리스트를 알 수 있다. 데몬을 발견한다면 마스터들의 리스트 파일을 찾거나, ICMP 메시지를 통해 마스터를 찾을 수 있다.

이상에서 살펴본 바와 같이, 탐지 및 추적을 위해서는 네트워크에 대한 지속적인 모니터링이 필요하다. 이를 통해, 공격명령을 포함한 패킷을 찾아내어 그 패킷을 보내거나 받는 시스템에 포함된 DDoS 도구를 찾고, 이를 바탕으로 추적을 수행하는 것이다.

2. 추적 시스템 제안

연구된 분석을 바탕으로 본 논문에서 제안하려는 추적 시스템은 다음과 같다. AS를 구성하는 각 라우터에서는 패킷 모니터링을 통해, 마스터와 데몬 간에 주고받는 통신패킷을 확인한다. 앞 절에서 분석한 특징을 갖는 패킷을 발견하게 된다면, 이 패킷의 목적지 주소를 통해 목적지 시스템을 알 수 있다. 이 패킷을 보낸 시스템은 마스터이거나 혹은 데몬일 수 있다. 마스터와 데몬간의 통신은 일대다 구조이기 때문에, 한 라우터를 통과하는 패킷을 모니터링한다면, 상대적으로 마스터가 보낸 패킷을 잡을 확률이 높다. 따라서 라우터 모니터링을 통해 마스터가 보내는 통신패킷을 검출해 데몬 시스템의 주소를 파악한다고 가정한다.

데몬 시스템을 찾은 후에는 데몬 시스템이 보내는 모든 패킷에 대해서 가장 가까운 라우터에서 IP 스푸핑 여부를 확인한다. 이 과정은 MAC 주소와 IP 주소를 비교함으로써 이루어진다. 데몬 시스템에서 발생한 정상적인 패킷이라면 IP주소를 변조하지 않을 것이며, 마스터에게 보내는 통신패킷일 경우에는 소스 IP를 변조한 패킷을 보낼 것이다. 따라서 이 패킷을 받는 목적지 시스템은 마스터로 의심할 필요가 있으며, 위조된 패킷을 받을 때마다 목적지 시스템에 대한 카운트를 증가시킨다. 이러한 정보를 각 라우터에서 수집하여 통계를 낸다. 각 라우터에서 위조된 소스IP를 갖는 패킷을 받았을 경우, 목적지 주소에 대한 카운트를 증가시켜 여러 라우터에서 같은 목적지에 대한 카운트가 보고된다면, 이 시스템을 마스터로 확인하게 된다.

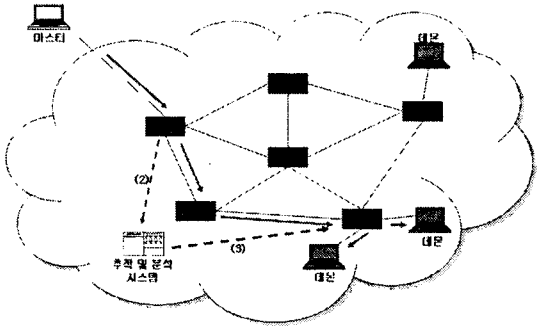

그림2: 추적 시스템 동작과정-1

1. 라우터에서의 모니터링을 통한 마스터->데몬으로의 패킷 검출
2. 추적 및 분석 시스템에 사실 통보
3. 데몬과 가장 가까운 라우터에 데몬 시스템이 보내는 모든 패킷에 대한 IP 스푸핑조사

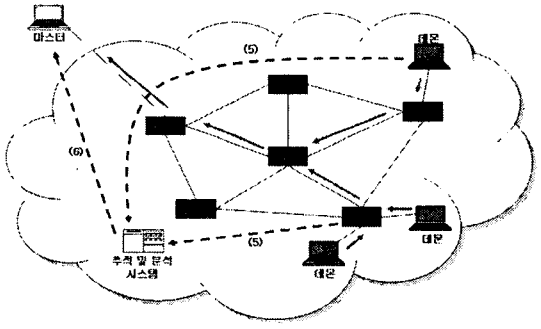

그림3: 추적 시스템 동작과정-2

4. IP스푸핑 조사를 통해, 변조된 패킷확인
5. 스푸핑 패킷의 목적지 주소에 대한 카운트 정보 수집
6. 카운트가 높은 시스템을 마스터로 확인

DDoS 도구에서 데몬이 자신의 IP를 변조하지 않고 마스터와 통신하는 경우도 있으며, 이를 위해 변조되지 않은 IP 패킷에 대해서도 앞 절에서

설명한 방법을 통해 통신패킷을 찾아 마스터를 확인하는 과정이 필요하다. 이 작업은 라우터에서 수행할 경우 과부하를 초래하므로, 별도의 추적 시스템에서 처리하도록 하는 것이 효율적이다.

라우터에서 수행하는 모니터링 과정을 흐름도로 나타내면 아래 그림과 같다.

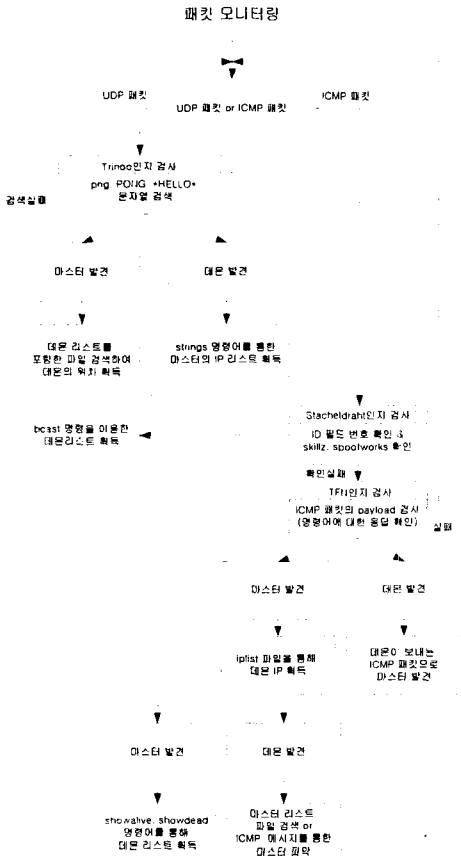

그림4 : 라우터 모니터링 흐름도

그림 4에서는 Trinoo, TFN, Stacheldraht의 3가지 공격을 탐지하고 핸들러, 에이전트를 추적하는 모듈로 구성되어 있다. 각각의 모듈은 공격 도구의 특징을 충분히 반영하였으며, 3가지 공격 도구 외에 차후 추가로 개발되는 공격 도구는 분석을 통하여 새로운 모듈로 추가할 수 있다. 즉, DDoS 공격 도구가 발견된 초기에 분석하여 모듈로 추가함으로써 사례별로 탐지 및 추적을 가능하게 할 수 있는 것이다. 제시한 시스템 모델은 침입탐지 시스템이나, DDoS 공격을 막는 시스템과 연동하여 동작하는 것을 목표로 한다.

IV. 결론 및 향후 연구방향

이상으로 DDoS 공격도구별 통신특성과 이에 기반한 탐지 및 추적방안에 대하여 살펴보았다. 본 논문에서 제안한 시스템은 패킷 모니터링을 근간으로 하고 있다. 전체 AS에 대한 관리권한을 갖고 있는 상황에서는 각각의 DDoS 도구에 대한 대응방안을 통해 마스터와 데몬을 추적하는 방안과, 관리권한이 없는 상황에서는 라우터에서의 IP 위조 패킷의 목적지 주소를 통해 추적하는 방안을 제시하고 있다. 향후 연구과제로는 다양한 DDoS 공격도구에 대해 심도깊은 분석을 통해 모니터링 시스템의 정확성 및 적용성을 높이는 연구가 필요하며, DDoS의 다른 계층에서의 추적방안과 연동하여, 보다 완전한 추적시스템을 제시하는 것이다.

참고문헌

[1] 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", Proceedings of the 1997 IEEE

[2] David Dittrich, "The DoS Project's trinoo distributed denial of service attack tool", October 21, 1999

[3] David Dittrich, "The Tribe Flood Network distributed denial of service attack tool", October 21, 1999

[4] David Dittrich, "The stacheldraht distributed denial of service attack tool", December 31, 1999