DDoS 도구 분석을 통한 공격 패킷 탐지 및 공격 네트워크 파악

김진혁[0], 홍만표
아주대학교 정보통신전문대학원
{forever[0], mphong}@ajou.ac.kr

Ajou University

Graduate School of Information and Communication

Jin-Hyok Kim[0], Man-Pyo Hong
Dept. of Information Engineering

요 약

최근에 인터넷을 통한 해킹이나 바이러스 침투로 인한 피해 사례들이 지속적으로 증가하고 있다. 2000년 2월, 야후, 아마존, CNN에 발생했던 DDoS(Distributed Denial of Service)[1, 2] 공격으로 인해 각 웹사이트들은 큰 피해를 입었던 사례가 있다. 야후의 경우 초당 수 기가 비트의 서비스 요청으로 인해 무려 3시간 이상 동안 서비스가 중지되는 사태까지 이르렀다. 이 사건은 분산 환경에서의 서비스 거부 공격의 위험성을 보여주고 있다. 본 논문에서는 지금까지 개발된 분산 서비스 공격 도구를 분석하고 이들이 사용하는 패킷을 탐지하여 공격을 위해 사용되는 경로를 파악하는 방법을 제안한다.

1. 서 론

현재의 인터넷 서비스들은 웹서버, 메일서버, DNS 서버 등과 같이 컴퓨터나 네트워크의 자원을 불특정 다수에 대하여 서비스를 제공해야 하므로 특정한 인증 수단 없이 인터넷에 공개되어 있는 요소가 많이 있는 상황이다. 보통 보안 인증 기법이나 암호화 기술을 가지고 접근제어를 하여 서버 보안을 하고 있지만, 웹서비스와 같이 불특정 다수에 대해 서비스하는 공개 서버에는 인증이나 암호화 기술을 적용하기가 힘들기 때문에 보안이 매우 취약하기 때문에 이러한 공개 서버가 공격 대상이 되는 경우가 많다. 예전에는 단순히 시스템에 침입하여 중요 자료를 빼가거나 다른 시스템에 침입하기 위한 중간 경유지로 사용하는 것이 주요 목적이었으나, 최근에는 고의적으로 시스템이나 네트워크를 마비 시켜 정상적인 기능을 수행하지 못하게 하여 사용자나 시스템에 피해를 주는 사례가 많아지고 있다. 후자의 경우가 바로 DoS 공격이라고 불리어지는 공격으로 방어하기 상당히 어려운 공격이다. 그러나 최근에는 이 공격법에 분산 네트워크 기법을 적용하여 보다 공격의 강도를 높이고 탐지하기도 어렵게 만드는 DDoS 공격법이 많은 피해를 주고 있다. DDoS 공격은 자동화된 도구를 이용하여 공격을 수행하기 때문에 누구나 쉽게 단시간 내에 수행할 수 있다. 그리고 여러 호스트를 활용하여 분산 네트워크를 구성하여 공격을 수행 하기 때문에 공격에 대한 피해자는 직접적으로 공격당한 Victim 호스트가 되겠지만, 자신도 모르게 중간의 공격 네트워크의 구성원인 Master나 Agent로 사용된 중간 호스트들도 피해자가 되는 것이다. DDoS 공격이 이러한 특성을 가지고 있기 때문에 Victim 호스트를 알아내어 미리 대책을 세우는 것도 중요하지만 중간 노드로 사용되는 호스트를 찾아내는 것도 중요하다. 결국 DDoS의 공격 네트워크를 파악하는 일이 중요하다고 할 수 있다. 공격 네트워크를 알아낸다는 것은 도구가 설치된 호스트가 얼마나 되는지 또 어느 호스트가 중간 경유지로 사용되는지에 대한 정보를 알아낸다는 것이다. 따라서 이러한 정보를 가지고 있다면 DDoS 공격에 대해서 Victim 뿐만 아니라 중간 노드로 사용되는 호스트에 대한 대책을 세울 수 있으며 공격의 강도도 약화시킬 수 있다.

본 논문에서는 DDoS 공격 도구에 대하여 분석과 분석을 통해 정의된 공격 패킷 탐지를 통해 대략적인 DDoS 공격 네트워크를 파악하는 방법에 대해 기술한다.

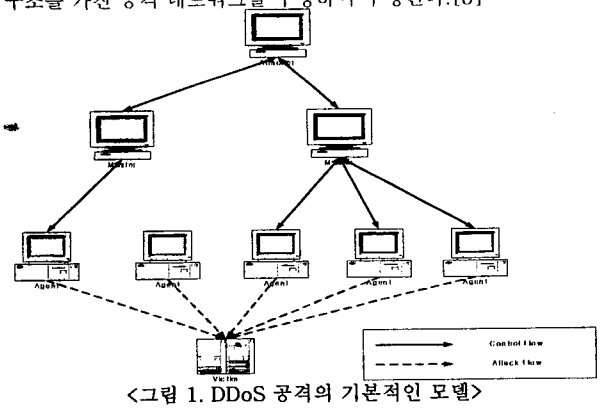
2. 본 론

2.1. 분산 서비스 거부 공격

분산 서비스 거부 공격이란 분산 환경에서 서비스 공격 도구들을 대규모 네트워크의 많은 호스트에 불법적으로 설치하고 이 설치된 도구들을 이용하여 서로 통합된 형태로 패킷을 범람시켜 심각한 네트워크 성능저하 및 시스템 마비를 유발하는 공격 방법을 말한다. 기존의 서비스 거부 공격에 분산 처리 개념이 도입되고 모든 공격이 자동화 되어 널리 유포된 DDoS 공격 도구를 이용하도록 되어 있어 누구나 쉽게 사용할 수 있도록 되어있다. Attacker는 우선 스니핑 도구나 스캐닝 도구를 이용하여 인터넷상의 취약성이 있는 호스트를 찾아내고 이들에 대해 침입을 시도하게 된다. 침입이 성공하게 되면 도구를 설치하고 도구는 다시 취약성 있는 호스트들을 찾아 자동으로 다시 다른 호스트에 도구를 설치하는 작업을 수행한다. 이러한 과정을 통해 수백, 수천대의 호스트에 도구를 설치할 수 있으며 자동화 되어있어 단 몇 십 분이면 이 모든 작업을 완료할 수 있다. 이후에 Attacker가 공격 명령을 보내게 되면 수백 대의 호스트가 공격 대상으로 지정한 Victim에 대해 DoS 공격을 수행하게 된다. 위에서 말한 것과 같이 분산 서비스 거부 공격에 사용되는 도구는 스캐닝, 백도어 설치, 암호화와 같은 지금까지 개발된 공격 기법의 대부분을 이용하며 거기에 분산화, 자동화, 에이전트화 같은 새로운 공격 기법도 사용하고 있다.

2.2. DDoS 공격 네트워크

현재 알려진 도구를 이용한 공격법은 모두 다음과 같은 기본적인 구조를 가진 공격 네트워크를 구성하여 수행된다.[3]



<그림 1. DDoS 공격의 기본적인 모델>

Attacker(or Clienr) : 모든 공격을 주도하는 공격자로서 DDoS 도구들을 원격으로 제어하고 직접 명령을 전달한다.
Master(or Handler) : Attacker로부터의 명령을 받아서 자신이 관리하는 Agent들에게 공격을 지시하는 역할을 한다.
Agent(or daemon) : Master에 의해 조정 받으며 공격 프로그램은 각각 Master로부터 받은 명령을 수행하며 최종적으로 Victim에게로 DOS공격을 수행하는 역할을 한다.
Victim (or Target) : 공격의 최종적인 피해자로 여러 호스트로부터 동시에 DoS 공격을 받게 된다.

이 공격 네트워크는 위의 크게 네 가지의 공격 노드로 구성된다. 결국 DDoS 공격 네트워크의 각 노드의 위치에 대한 정보가 DDoS 공격을 판단하는 가장 중요한 정보가 된다.

2.3. DDoS 공격 도구의 종류, 기능 및 대응 방안

현재 사용되는 공격 도구에는 Trinoo, Stacheldraht, TFN, TFN2K, Mstream, Shaft 등이 있다. 이 도구들이 가지고 있는 기본적인 기능은 시스템에 대해 취약성을 검사한 다음 그 취약성을 이용하여 접근하여 Master또는 Agent 도구를 대상 시스템 내에 설치하는 기능이다. 주로 Linux나 Unix 시스템들에 대해 이 과정을 수행하는데, 대부분 RPC Buffer Overflow 취약점으로 직접적인 공격을 수행하여 쉽게 root권한을 얻으며, 이후에는 Root Kit과 같은 도구를 사용하여 침입 사실을 숨기는 작업도 같이 수행한다.
DDoS 공격의 경우에도 바이러스나 웜과 마찬가지로 근본적으로 막을 수 있는 방법은 없다. 현재 알려진 이 공격에 대한 대응 방안으로는 DDoS 공격 도구가 알려진 취약점을 이용하여 다수의 호스트를 장악하고 여기에 공격 도구를 설치하는 과정을 수행한다는 특성을 파악하여 일단 호스트의 취약성을 검사하고 패치를 이용해 취약성을 제거하는 하는 방법이나 파이어월이나 라우터 단에서 비정상 패킷을 필터링하는 방법 정도이다.[4, 5] 근본적인 방어책이 없는 만큼 공격의 조기 탐지와 추적 및 탐지 시의 즉각적인 대응이 그만큼 중요하다.

2.4. DDoS 공격 도구 분석

DDoS 도구 소스의 직접적인 분석과 기존의 도구들에 대해 분석되어 있는 자료[6]를 참고하여 다음의 표를 구성하였다.

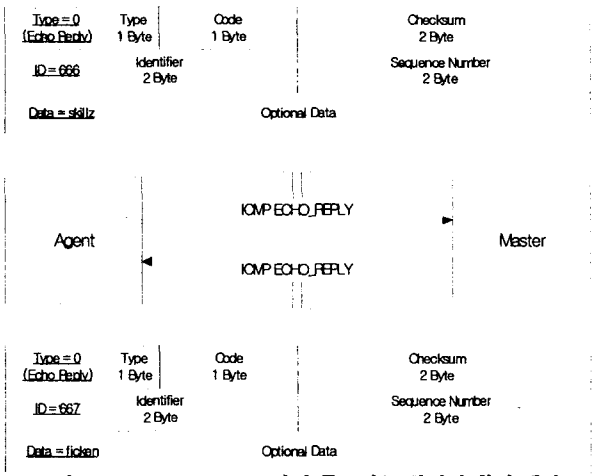
<표 1. 4 가지의 대표적인 DDoS 공격 도구의 분석 비교>

	Trinoo	Stacheldraht	TFN	TFN2K
DoS 공격 기법(Agent의 기능)	UDP floods	UDP/SYN/ICMP floods, Smurf	UDP/SYN/ICMP floods, Smurf	UDP/SYN/ICMP floods, Smurf
Attacker ↔ Master	27665/tcp	16660/tcp (암호화)	TELNET을 이용	TELNET을 이용
Master ↔ Agent	27444/udp	ICMP echo reply, 65000/tcp	ICMP echo reply	UDP/TCP/ICMP를 Random하게 사용(암호화)
Agent ↔ Master	31335/udp	ICMP echo reply	ICMP echo reply	없음
통신암호화	X	O	X	O
IP spoofing	X	O	O	O
PS 숨기기	X	O	O	O
제작시기	1999년 이전	1999년 8월	1999년 이전	1999년 11월

위 표를 살펴보면, 기본적인 기능 외에도 통신 암호화, IP 스푸핑, 프로세스 숨기기 등 탐지하기 어렵게 하기위한 여러 가지 기술이 추가 되어있는 것을 볼 수 있다. 통신 암호화를 사용하게 되면 통신 패킷을 감시한다고 해도 탐지하기 쉽지 않으며, 스푸핑된 IP를 사용하게 되면 공격 패킷이 생성된 실제 위치를 파악하는데도 많은 어려움이 생긴다.

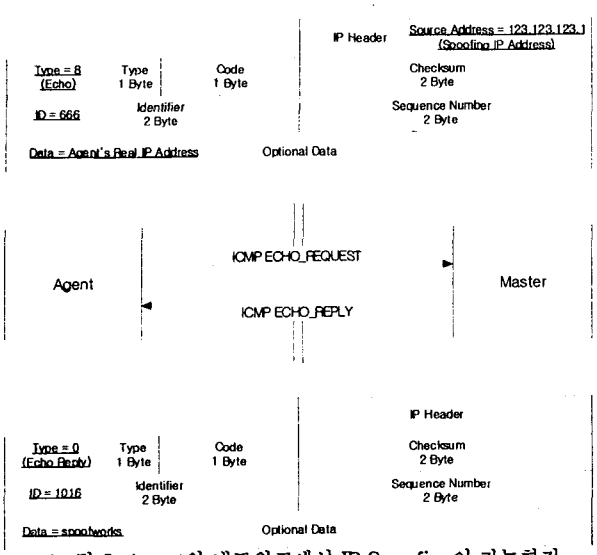
2.5. DDoS 공격 도구 분석을 통한 패킷 탐지

위 표를 살펴보면 Trinoo나 TFN의 경우에는 제작된 시기가 비교적 오래 전이기에 통신 시에 암호화 기능을 사용하지 않는다는 것을 알 수 있다. 이 경우에는 도구간의 정보 교환이나 공격 명령 전달을 위해 사용하는 패킷을 분석하게 되면 각 패킷마다 도구의 기능을 수행하기 위한 특정 문자열을 포함하거나 헤더 정보 등을 변환하는 것을 알 수 있다. 따라서 도구에서 사용하는 패킷의 패턴을 비교적 쉽게 발견할 수 있다.
암호화된 통신을 사용하는 Stacheldraht나 TFN2K의 경우에도 모든 경우에 암호화된 패킷을 이용하여 통신을 하는 것은 아니다. Stacheldraht의 경우에는 Master가 Agent를 인식하는 과정에서 해당 호스트에서 도구가 현재 실행중인지를 검사하는 패킷을 보낸 후 응답을 보고 연결하는 패턴을 가진다. 따라서 Master ↔ Agent간의 연결에는 암호화가 사용되지 않으므로 암호화를 사용하지 않는 도구의 패킷 패턴을 찾는 것과 마찬가지로 패킷의 특성만 파악한다면 쉽게 탐지할 수 있다. Master ↔ Agent간의 연결 확립을 위한 서로 활성화 확인 패킷을 주고받는 과정은 다음과 같다. 먼저 Agent가 실행될 때 자신의 Master들에게 ICMP_ECHOREPLY 패킷의 ID 필드를 666으로 설정하고 DATA 필드에 "skillz"라는 단어를 넣어서 보낸다. 자신이 동작중이라는 것을 알리기 위한 기능이다. 그러면 이 패킷을 받은 Master 프로그램은 다시 ICMP_ECHOREPLY 패킷의 ID필드에 667을 설정하고 DATA 필드에 "ficken"이라는 단어를 넣어서 응답을 하게 된다. 이는 Master가 Agent가 보내준 신호를 받았다는 메시지를 받았다는 일종의 acknowledgment 기능을 하는 패킷이다.[7]



<그림 2. Master ↔ Agent 간에 주고받는 활성화 확인 패킷>

또 한 가지 이러한 과정에서는 반드시 해당 호스트로부터 응답을 받아야만 Master, Agent를 인식을 할 수 있기 때문에 위조된 아이피가 사용될 수 없다. 결국 패킷을 탐지한다면 Master와 Agent의 위치를 파악 할 수 있게 되는 것이다. 이것은 현재 어느 호스트에 DDoS 공격을 위한 어떠한 유형의 도구가 설치되어 있는지 파악하는데 도움을 줄 수 있다. 예를 들면 "아이피 123.123.123.1 이라는 호스트에 Stacheldraht 도구의 Master가 설치되어 있으며, 아이피 123.123.123.2 이라는 호스트에는 Stacheldraht 도구의 Agent가 설치되어 있다"와 같은 정보를 알 수는 것이다.
이와 비슷한 유형으로 Agent의 네트워크에서 IP Spoofing이 가능한지 검사하는 과정을 이용할 수도 있다. 특정한 네트워크의 라우터에서는 자신의 네트워크에 포함되지 않은 IP주소를 source IP로 갖는 IP 패킷을 통과시키지 않는다. 즉 IP Spoofing을 허용하지 않는 경우가 있다. 이런 경우에는 Agent프로그램은 Victim에 대한 DoS 공격 시 IP Spoofing기능을 사용할 수 없게 되기 때문에 미리 이에 대한 검사를 수행하는 작업을 한다. Agent가 실행되면, 자신의 Master로 IP 헤더 부분을 소스 아이피는 123.123.123.1, ICMP 타입은 7, 코드는 0, ID는 666으로 설정하고 DATA에 Agent의 실제 아이피를 넣어서 보낸다.
이 패킷을 받은 Master는 ICMP DATA필드를 통해서 Agent의 실제 IP를 알아내고 자신이 관리할 Agent리스트에 등록하고, 다시, Agent에게 ICMP_ECHOREPLY 패킷에 ID를 1016로 설정하고 DATA에 "spoofworks" 라는 단어를 넣어서 Agent로 보낸다. Agent에서 ICMP_ECHOREPLY패킷을 받으면 자신의 네트워크가 IP Spoofing이 가능하다는 것을 알게되고 이후에 Spoofing된 아이피를 사용하여 공격을 하게된다.[7]



<그림 3. Agent의 네트워크에서 IP Spoofing이 가능한지 검사하는 과정>

이 기능을 수행하는 패킷을 탐지 하게 되면 마찬가지로 도구 설치 호스트의 위치에 대한 정보를 얻을 수 있다. 이외에도 명령 전달 패킷을 탐지하여 호스트의 위치에 대한 정보를 얻을 수 있다.

2.6. 도구의 공격 패킷을 역이용한 탐지 방법

DDoS 공격 도구를 분석하여 생성되는 패킷의 유형을 알아낸다면 그 패킷을 역이용하여 도구의 설치 상황을 탐지하는 방법을 생각해볼 수 있다. 도구가 어떤 기능을 수행하기 위해 생성하는 패킷을 그대로 모방해서 생성한 다음 그 패킷을 도구가 설치되었을 것으로 짐작되는 호스트로 보내보고 그 응답을 봄으로써 해당 호스트에 실제로 도구가 설치되었는지의 여부를 판단할 수 있다. 탐지 에이전트가 특정 호스트에 대해 Stacheldraht가 주로 사용하는 포트로 암호화된 패킷이 지나다니는 것을 발견 하였다고 가정하자. 그렇지만 이를 가지고 Stacheldraht 공격 도구에 의해 생성된 패킷이 지나 다닌다고 확신 할 수는 없다. 이때 위에서 예로 제시한 Stacheldraht의 Master ↔ Agent간에 주고받는 활성화 확인 패킷을 모방하여 도구가 사용하는 방법을 역이용하도록 한다. 도구가 활성화 확인을 위해 사용하는 "skillz"라는 단어를 포함하는 패킷을 생성하여 의심이 가는 호스트에 보낸 후 그 응답을 기다린다. 의심받는 호스트에서는 이 패킷을 받았을 경우에 만약 Stacheldraht 도구가 설치되어 있다면 이 패킷이 정상적인 Master에서 생성된 패킷이라고 여기고 "ficken" 이라는 단어를 포함하는 패킷을 그에 대한 응답으로 보낼 것이다. 그렇다면 응답을 기다리는 호스트에서는 "ficken"이라는 단어를 포함한 패킷이 응답으로 온 것을 확인하게 될 것이다. 결국 의심 하고 있던 호스트에 Stacheldraht 도구가 설치되어있다고 확신할 수 있고 이에 대한 조치를 취할 수 있을 것이다.
마찬가지로 IP Spoofing이 가능한지를 검사하는 패킷을 이용할 수도 있으며, 이들과 비슷하게 서로 패킷을 보내고 그 패킷에 대한 응답을 통해 정보를 전달받는 구조로 되어있는 경우는 모두 이용 가능하다.

2.7. 공격 네트워크 파악

암호화를 사용하지 않는 도구의 경우에는 도구 분석을 통해 공격을 위해 생성된 패킷을 정의 할 수 있으며, 암호화 되어 통신하는 도구의 경우에는 위에서 예를 든 것과 같은 암호화를 사용하지 않는 기능이나 암호화된 패킷을 이용할 수 없는 기능을 수행하는 패킷에 대한 탐지를 이용하여 도구 설치에 대한 정보를 얻을 수 있다. 이러한 정보를 이용하여 최종적으로 공격을 위해 도구들로 구성된 대략적 공격 네트워크 구성을 파악하는 데에 사용하고자 한다. 일단 공격 네트워크를 파악 하게 된다면 Victim에 대한 직접적인 공격이 수행되기 이전에 도구가 설치된 호스트에 경고를 줌으로서 도구를 제거하거나 중요한 거점이 되는 호스트에 설치된 도구를 무력화 시켜 최종 공격이 수행될 시에 공격 효과 감소에 큰 도움을 줄 수 있다. 우선 이러한 공격 네트워크를 파악 하기위해서는 DDoS 공격 도구에 의해 생성된 패킷을 탐지할 수 있는 탐지 에이전트가 있어야 하며 이것을 호스트에 설치하여 감시하도록 하고 각 에이전트 간에 서로 정보 공유를 하도록 구성해야 할 것이다.

2.8. 구현

위에서 설명한 것과 같은 방법으로 도구 분석을 통해 도구가 사용하면 공격용 패킷의 패턴을 정의 하였다. 그리고 정의된 패턴 중에서 활성화 확인 패킷과 같은 도구의 위치 정보를 알 수 있는 패턴을 다시 정의 하였다. 기본적인 구현에서는 이렇게 정의된 패턴을 사용하였다. 구현된 모듈의 기본적인 기능을 살펴보면 우선 호스트에 일종의 데몬 형태로 실행 중이며 탐지 에이전트 역할을 수행한다. 패턴 데이터베이스에 탐지할 패킷의 패턴을 도구 별로 정의한다. 그리고 이 정의된 패턴의 패킷이 호스트에서 탐지될 경우 해당 패킷을 분석하여 무슨 기능을 하는 패킷인지 어디로부터 온 패킷인지 등에 관한 정보를 가지고 현재 자신의 호스트에 도구의 Master나 Agent가 설치되었을 가능성이 있는지 다른 어느 호스트에 도구의 Master나 Agent가 설치 되어있는지를 판단하게 된다. 이러한 판단이 내려지게 되면 도구가 설치되었을 것으로 의심이 되는 호스트들에 대한 리스트를 생성한다. 그리고 의심이 되는 호스트에 대해 위에서 제시한 방법과 같이 도구가 사용하는 패킷을 역이용하여 확신을 하는 단계를 거치게 된다. 이러한 과정을 거쳐 최종적으로 도구가 설치되어있는 호스트들을 판별하게 된다.
현재 구현된 탐지 에이전트 모듈의 기본적인 기능은 비교적 간단하지만, 이 탐지 에이전트를 다른 호스트에 분산 설치하고 서로 도구 설치 리스트나 필요한 정보를 교환하며 일종의 탐지 에이전트만의 네트워크를 형성하게 된다면, 즉 탐지 에이전트가 극소수가 아닌 어느 정도의 수의 호스트에 설치되었다고 가정한다면, 비록 DDoS 공격에 대한 전체 공격네트워크는 알 수 없을 지라도 대략적인 구조는 파악할 수 있을 것이다.

3. 결 론

현재 DDoS 공격은 기존의 공격 기법과는 달리 분산화, 에이전트화 됨으로써 많은 피해를 주고 있다. 또 이러한 기능을 수행하는 도구들은 앞으로도 더욱 지능화되고, 더 많은 고급 기능을 추가해감으로써 탐지하기가 쉽지 않을 것이다. 본 논문에서는 현재까지 DDoS 공격에 사용되는 도구들을 분석하고 DDoS 공격 네트워크의 모습을 알아내어 공격 피해를 감소시킬 수 있는 방법을 제안 하였다. 도구 분석을 통해 사용되는 공격 패킷의 패턴을 알아내고 이러한 패킷을 탐지할 수 있는 탐지 에이전트를 이용하여 DDoS 공격을 수행하는 경로로 사용되는 Master나 Agent의 위치를 알아내는 방법을 이용하였다. 이런 유형의 방법이 공격 네트워크 파악을 위한 기본적인 방법이 되며, 향후 세부적인 도구 분석, 공격 패킷 탐지 알고리즘 추가하여 탐지 모듈을 가진 에이전트로 발전시키고 에이전트 간의 정보 교환에 대한 연구가 지속된다면 지금보다 좀 더 구체적이며 효과적으로 DDoS 공격 네트워크에 대한 구조를 파악할 수 있을 것이며, 이를 이용하여 DDoS 공격에 대해 좀더 근본적으로 대응할 수 있는 방법을 찾을 수 있을 것이다.

IV. 참고문헌

[1] Felix Lau, Stuart H. Rubin, Michael H. Smith, Ljiljana Trajkovic, "Distributed Denial of Service Attacks", Systems, Man, and Cybernetics, 2000 IEEE International Conference on, Volume: 3, 2000
[2] Xianjun Geng and Andrew B. Whinston, "Defeating Distributed Denial of Service Attacks", IT Professional, Volume: 2 Issue: 4, July-Aug. 2000
[3] CERT/CC(Computer Emergency Response Team/Coordination Center), "Distributed Intruder Tools Workshop Report", 1999. 11.
[4] http://www.sans.org/dosstep/index.htm, "Help Defeat Denial of Service Attacks : Step-by-Step" 2000. 5.
[5] http://www.cisco.com/warp/public/707/newsflash.html, "Strategies to Protect Againdt Distributed Denial of Service Attack" Cisco System, 2000. 2.
[6] http://staff.washington.edu/dittrich/misc/ddos/, "Analyses and talks on attack tools" 1999 - 2000
[7]http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt, "The "stacheldraht" distributed denial of service attack tool" 1999. 12.