파일 공유를 위한 P2P 어플리케이션 구조와 보안 위협

김봉한 (청주대학교)

차 례

1. 서론

최근에 인터넷 비즈니스에서 많이 논의되고 있는 P2P의 용어는 Peer-to-Peer의 줄임말로서 현재 활발히 진행되고 있는 인터넷 비즈니스의 중요한 변화를 표현하는 함축적인 용어이다. 이것은 기존의 인터넷 비즈니스의 지배적 구조였던 클라이언트-서버 중심의 비즈니스 모델에서 P2P 구조로의 변화한다는 의미이다. P2P는 인터넷에서 중간에 서버 컴퓨터를 거치지 않고 정보를 찾는 사람과 정보를 가지고 있는 사람의 PC를 직접 연결시켜 데이터를 공유 할 수 있게 해주는 기술과 그 기술을 응용해 만든 새로운 서비스를 말한다. 인터넷상의 정보를 검색 엔진을 거쳐 찾아야 하는 기존 방식과는 달리 인터넷에 연결된 모든 개인 컴퓨터로부터 직접 정보를 제공받고, 검색은 물론 다운로드까지 할 수 있다.

근거리통신망(LAN)을 인터넷으로 확대한 개념으로 이 기술을 활용하면 PC 사용자가 별도의 서버나 고정IP(전용선) 없이도 인터넷으로 서로의 컴퓨터를 자유롭게 드나들며 필요한 자료를 주고받을 수 있기 때문에 일반 PC 사용자는 MP3 파일, 동영상, 컴퓨터 파일을 중간 매개자 없이 직접 주고받을 수 있으며 이를 인터넷 비즈니스에 접목하면 홈 네트워킹, 중소기업 애플리케이션 서비스 제공(ASP) 사업 등에 폭넓게 적용할 수 있다[1].

실제로, P2P는 기존의 클라이언트-서버에서 발생할 수 있는 고정IP 부족 현상을 가변IP를 사용함으로서 어느 정도 해소할 수 있고 동시에 통신망사업자의 고정IP 주소관리에 따른 부담도 획기적으로 줄여줄 수 있고, 이 서비스는 초고속통신망, 케이블망을 이용한 일반 가정용 전자상거래 망 구축은 물론 네트워크 환경이 열악한 일반 중소기업의 정보화사업 추진에도 유용하게 활용되고 있다.

본 고에서는 인터넷 서비스 트래픽중 가장 많은 트래픽 양을 차지하고 있는 P2P 어플리케이션의 구조와 P2P에서 발생할 수 있는 공격, 그리고 P2P 서비스에 구현되어야할 정보보호 서비스에 대해 고찰하고자 한다.

2. P2P 어플리케이션 구성

2.1 구성요소

P2P 어플리케이션의 구성요소들은 어플리케이션의 운영을 원활하게 수행하기 위해 특정한 역할을 수행한다. 구성요소에 대한 각각의 범위와 역할은 다음과 같다[2].

가. 리스너

리스너가 수행하는 첫 번째 기능은 서버에 로그인하는 것이다. 이는 서버에게 리스너 자신의 존재유무를 알리고 자신의 공유된 자원들을 목록으로 만드는 것이다. 로그인 후 리스너는 서버로부터 넘겨받은 클라이언트들의 요청을 다룰 수 있다. 리스너는 클라이언트가 파일에 접근하면 언제나 그 파일을 다운로드해 준다. 다운로드 과정과 함께 리스너는 다수의 클라이언트의 요청도 쉽게 처리 할 수 있다. 리스너는 다수의 클라이언트를 다룰 때 본질적으로 서버처럼 작동하여 사용자가 검색 옵션을 사용할 수 있도록 해준다.

서버는 오직 루트 수준의 검색(공유된 자원들을 기본적인 수준에서 검색하는 것) 만을 수행한다. 루트 수준을 넘어선 검색을 하기 위해서는 리스너의 기능이 필요하다.

만약 리스너가 어떤 폴더를 공유된 자원으로 직접 선언한다면 사용자는 쉽게 그 폴더를 열어서 모든 파일의 목

록을 볼 수 있다. 이러한 검색은 서버 수준에서 수행하는 루트 검색보다 훨씬 빠르다. 리스너 수준에서의 검색은 콘텐츠의 위치에 관해 미리 정보를 가지고 단일 컴퓨터 상에서 일어난다. 리스너의 한가지 중요한 특징은 리스너가 사용자가 존재하지 않는 비사용자모드(Unattended mode)에서 실행 가능하다는 것이다.

나. 서버

서버는 모든 등록된 리스너들의 자원들에 관한 상세한 정보와 목록을 보유하고 있다. 이 자원들은 리스너들이 다른 피어들과 공유하기 위해 제공하는 것이다. 클라이언트 또는 리스너가 실행되는 컴퓨터상의 파일은 서버로 전송되는 것이 아니다. 리스너들의 목록을 제공한 후 서버는 리스너와 클라이언트 사이에 접속을 시작한다. 나머지 작업은 리스너와 클라이언트의 역할이다. 만약 클라이언트가 리스너가 실행되는 컴퓨터로부터 파일을 다운로드하기를 원하면, 서버와 관계없이 바로 다운로드할 수 있다. 클라이언트는 서버에게 리스너들의 목록을 보여줄 것을 요청할 수 있는데 이 목록의 리스너들은 클라이언트의 요청을 만족시킬 수 있는 것이다.

P2P 환경에서는 이러한 검색을 전역 요청(global request)이라고 부른다. 서버를 사용할 경우에 이런 검색과 관련된 또 다른 용어로는 루트 수준 검색(root search)이 있는데, 이는 리스너가 자신의 공유된 자원들을 선언할 때에 일어나는 검색이다. 간단히 말해서, 리스너와 클라이언트 사이에 접속을 수립한 후 서버는 클라이언트가 요청한 콘텐츠를 포함하고 있는 폴더 또는 디렉토리를 가려낸다. 이때 모든 폴더 또는 디렉토리에 관하여 심도 있는 검색을 수행하지 않는다.

다. 브라우저

브라우저는 사용자와 컴퓨터 사이에 인터페이스 역할을 한다. 클라이언트는 브라우저를 통하여 요청을 전송하고 리스너의 응답을 수신한다. 리스너와 달리 브라우저는 항상 사용자가 존재하는 모드(attended mode)에서 실행된다. 브라우저는 로그인한 리스너들의 목록을 보여주고 모든 요청을 리스너에게 전송한다. 이때 리스너는 스스로 모든 처리를 수행한다. 리스너가 요청을 처리한 후, 브라우저는 최종 사용자에게 결과를 보여준다. 모든 프로세스는 브라우저가 아니라 리스너가 수행한다는 것이다. 사용자는 오직 질의의 결과만을 볼 수 있을 뿐이다. 콘텐츠를 처리하는 것은 사용자들에게 보여지지 않고 숨겨져 있다.

브라우저의 도움을 받아 다양한 조건으로 공유된 자원을 검색할 수 있다. 예를 들어, 클라이언트는 리스너가 오직 문서 파일들만 보여주도록 요청할 수 있으며 또는 오직 실행파일들만 보여주도록 요청할 수 도 있다. 몇몇 조건에 근거한 검색과 함께 이용 가능한 리스너들의 목록을 얻기 위해 서버에게 전역 검색을 요청할 수 도 있다.

2.2 구성 요소들의 관계

서버와 리스너 그리고 브라우저는 각각의 임무를 수행하는 것과 별도로 서로 관계를 가지고 있다. 어플리케이션의 P2P 구조에 기반 하여 세 개의 구성 요소들의 관계는 다음과 같다[2].

가. 리스너-서버 관계

서버는 모든 리스너들에 대하여 이름, IP 주소, 그리고 가장 중요한 항목인 공유된 자원에 대한 정보를 데이터베이스에 보유하고 있다. 리스너는 서버에 로그인해서 공유된 자원을 선언한 후, 파일과 폴더의 이름을(파일과 폴더의 내용이 아니라) 위치와 함께 전송한다. 만약 모든 파일과 폴더를 내용과 함께 처리한다면 서버가 병목 현상을 일으키기 때문이다. 더구나, 파일 공유와 다운로드 때문에 발생하는 전체적인 부하는 서버의 부담이 되고 성능을 저하시킬 것이다. 또 하나 중요한 점은 서버가 제공하는 리스너들의 목록은 로그아웃 상태의 리스너들은 제외하며 현재 온라인 상태인 리스너들만을 보여준다.

나. 리스너-브라우저 관계

리스너와 브라우저의 관계는 상당 부분 브라우저가 모든 요청을 리스너에게 보내고 처리된 질의나 메시지의 형태로 응답을 수신하는 요청/응답 관계라고 말할 수 있다. 이 관계에서, 리스너는 리스너 수준의 모든 요청을 처리하고 브라우저는 처리된 결과를 클라이언트에게 보여준다. 클라이언트는 다음과 같은 다양한 형태로 요청을 보낼 수 있다.

- 클라이언트는 공유된 모든 파일 및 폴더를 보여줄 수 있다.
- 클라이언트는 리스너를 검색하거나 리스너에게 파일/폴더의 내용을 보여줄 것을 요구할 수 있다. 이것은

서버 수준에서는 일어날 수 없다.

- 클라이언트는 다운로드 요청을 리스너에게 넘기는 방법으로 콘텐츠를 다운로드 할 수 있다.
- 클라이언트는 자신의 파일을 리스너의 계정에 업로드할 수 있다.
- 클라이언트가 리스너로부터 파일을 다운로드/업로드하는 경우, 리스너가 클라이언트보다 더 우위에 있다. 왜냐하면 클라이언트와 콘텐츠를 공유할 때, 리스너가 자신의 콘텐츠나 정보를 읽기 전용, 쓰기 가능, 읽기 가능 상태로 지정할 수 있는 권한이 있기 때문이다. 이런 권한은 폴더에 속성으로 지정될 수 있다. 리스너는 이 권한을 사용하여 악의적인 의도를 가진 클라이언트가 중요한 정보를 다운로드 하는 것은 물론, 불필요한 파일을 업로드 하는 것을 제한할 수 있다.

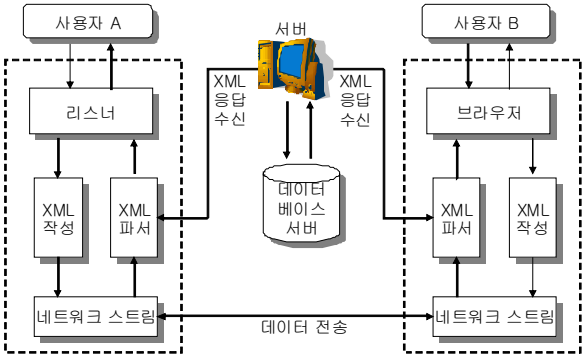
다. 브라우저-서버 관계

브라우저는 리스너들의 목록을 서버로부터 가지고 와서 어느 리스너가 자원을 공유하고 있는지 확인한다. 브라우저는 서버로부터 리스너들의 목록을 얻는 것과 함께, 루트 수준의 검색 요청을 전달 할 수 있다. 다르게 표현하면 브라우저는 원하는 콘텐츠에 대해 모든 리스너들을 검색할 수 있고 클라이언트의 요청이 있을 때, 클라이언트에게 모든 이용 가능한 리스너들에 관한 정보를 알려 줄 수 있다.

3. P2P 어플리케이션의 구조

일반적인 P2P 애플리케이션의 구성요소는 리스너, 브라우저, 그리고 서버 3개의 모듈에 기반을 두고 있다. 그림 1은 원격지에서 실행되는 두 개의 피어(리스너와 브라우저)가 서로 자원의 공유를 시도하는 것을 보여주고 있다. 두 가지 기본적인 동작을 수행한다.

- 요청 : 피어 사이에 계속적으로 수행할 상호 작용을 위해 통신을 수립하는 프로세스이다.
- 응답 : 요청에 대한 적절한 응답을 돌려주는 프로세스이다. 여기서 응답은 미리 정의된 메시지를 반환하는 것일 수도 있고 요청에 대한 단순한 응답일 수 도 있다.



▶▶ 그림 1. P2P 어플리케이션 기본 구조

이 어플리케이션에서 리스너로 작동되는 피어는 다른 피어들에 의해 만들어진 요청에 대하여 응답하는 역할을 하는 반면, 브라우저는 모든 요청을 생성하는 피어이다. 서버는 현재 로그인한 모든 피어들의 정보(IP 주소와 로그인 이름, 피어가 다른 피어들을 위해 공유한 자원들에 관한 정보)를 보유하고 있는 데이터베이스를 관리하고 있다. 이 어플리케이션은 많은 부분을 XML에 의존하고 있는데, 이는 배후의 컴포넌트들 사이에 일어나는 모든 통신이 XML 문서를 통해서 이루어지기 때문이다. 이 애플리케이션을 설계할 때 통신 수단으로 XML을 선택한 이유는 거의 모든 현대 프로그래밍 언어들이 XML을 지원하기 있기 때문이다.

P2P 어플리케이션은 XML을 처리하는 두 개의 컴포넌트를 별도로 제작하여 이용하고 있다.

- XML 파서 컴포넌트 : 생성된 요청과 요청에 대한 응답을 파싱하는데 이용된다.
- XML 작성 컴포넌트 : 적절한 요청과 응답을 생성하는데 이용된다.

어플리케이션이 시작되면 리스너는 자신의 로그인 이름, IP 주소, 공유된 자원들에 대한 정보로 구성된 HTTP 요청에 서버에 접속한 리스너들의 목록에 자신의 항목을 추가하고 서버는 리스너로부터 넘겨받은 사용자 정보의 인증 작업을 수행한다. 사용자 정보가 올바르면 서버에 의한 인증은 성공적으로 수행되는 반면, 사용자 정보가 올바르지 않으면 인증은 실패한다. 서버는 모두 XML 형태로 적절한 응답을 반환하는데, 이 응답은 XML 파서로 전달된 후 파싱된다. 최종적으로 적절한 메시지가 리스너에 표시된다.

이제 리스너 역할을 하는 피어가 서버에 성공적으로 로

그인했다고 가정 할 수 있으며 이 리스너의 요청에 대한 XML 응답은 인증 작업을 거친 후 에만 반환된다. 하지만 브라우저의 요청에 대해서는 서버가 자신에게 접속한 모든 피어들의 목록을 XML 형태로 바로 반환해 준다. 서버는 접속된 리스너들의 이름, 공유된 자원, 리스너의 IP 주소로 구성된 목록을 브라우저에게 제공한다.

이제 브라우저는 반환된 리스너 목록 중에서 하나의 리스너를 선택할 수 있다. 여기서부터 서버의 역할은 없어진다. 왜냐하면, 서버는 피어들 사이의 통신을 시작해 주는 자신의 임무를 모두 완수했기 때문이다. 서버는 통신을 시작할 수 있도록 정보를 제공해 주는 역할을 할 뿐이며, 피어(리스너)와 통신하기 위해서 수립하는 것은 브라우저이다. 리스너는 브라우저와 통신을 개시하는 것이 아니라 단지 브라우저들에 의해 접속 요청을 검색한다.

즉 브라우저는 서버에 접근한 후 원하는 리스너와 통신을 수립하는 두 단계를 수행한다. 일단 브라우저와 리스너 사이에 통신이 수립되면 두 피어들은 서로 자유롭게 데이터를 주고받을 수 있다. 브라우저는 XML 형식으로 데이터를 읽고 쓰기 위한 네트워크 스트림을 개방한다. 피어들 사이에 일어나는 주요 프로세스는 이 네트워크 스트림을 통한 데이터 업로드와 다운로드이다.

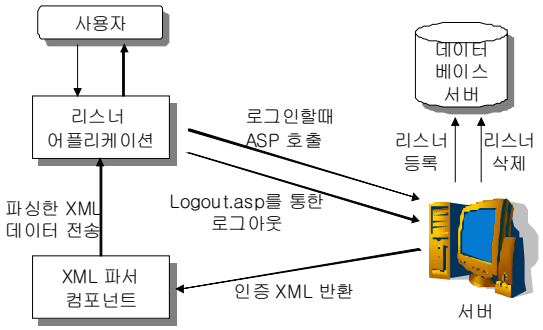
브라우저는 특정 데이터를 원격 리스너로부터 다운로드 하는 방법으로 원하는 콘텐츠를 얻는다. 브라우저가 데이터를 다운로드 하려면 XML 요청을 사용한다. 차례로 리스너는 자신에게 수신되는 요청 XML을 파싱하고 최종적으로 어떤 파일이 브라우저에게 전송할 것인지 판단한다. 이 정보는 브라우저의 요청에 대한 XML 응답이 될 수 있고 브라우저가 다운로드 요청한 파일일 수도 있는데, 이 정보를 브라우저의 네트워크 스트림에 전달함으로써 브라우저에게 전송된다. 최종적으로 브라우저는 리스너에 의해 업로드된 파일을 읽어들이고 이때 읽어들인 파일이 브라우저의 요청에 대한 응답인지 다운로드 요청을 한 파일인지를 확인한다. 그리고 적절한 메시지를 화면에 보여준다.

브라우저가 리스너로부터 파일을 다운로드 하는 것과 동일한 방법으로 브라우저는 리스너에게 파일을 업로드 할 수 있다. 이를 위해 브라우저는 다른 피어들이 공유한 메모리 영역을 선택하고 자신이 업로드 할 파일을 선택한다. 그리고 리스너에게 보낼 요청 XML을 생성한다. 리스너가 요청 XML을 수신하면, 작성되어야 할 파일에 대한 충분한 권한이 있는지 확인하기 위하여 폴더의 속성을 점검한다. 데이터를 쓰기 위한 충분한 권한이 있다면 브라우저가 쓰고 있는 파일을 읽어 들이고 동시에 데이터를 파일에 쓴다. 쓰기 권한이 없을 경우, 업로드 거부 응답을 생성하여 브라우저에게 반환한다.

3.1 ASP를 사용한 리스너와 서버 사이의 통신

리스너 어플리케이션을 시작하면 로그인 창이 나타나며, 사용자는 이 로그인 창에 로그인 이름을 입력한다. 실제로 어플리케이션 내부에서 통신에 이용하는 정보는 로그인 이름이 아니라 리스너 IP 주소이다. 사용자가 로그인 이름을 입력한 후 로그인 버튼을 누르면 리스너는 login.asp라는 이름의 ASP 페이지를 호출한다. 리스너는 그림 2와 같이 서버에 자신의 정보를 등록하기 위하여, 공유된 자원들의 목록과 로그인 이름, 자신의 IP 주소를 전달하여야 한다.

서버는 넘겨받은 모든 인수를 확인 한 후 해당 리스너의 항목을 작성한다. 그리고 요청에 대한 응답으로 리스너에게 XML을 반환한다. 이 후 서버는 사용자가 로그아웃할 때까지 리스너와 통신하는 일은 없다. 사용자가 로그아웃할 때, 리스너는 서버에 등록된 자신의 정보를 삭제하라는 요청을 ASP 페이지를 통해 전송한다.



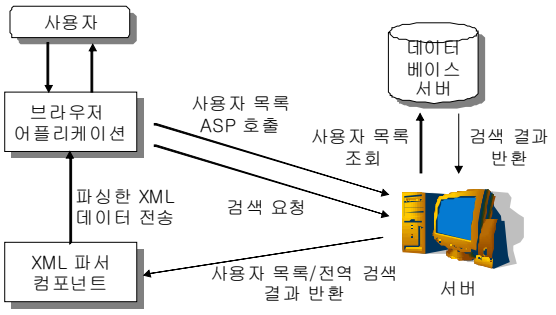
▶▶ 그림 2. 리스너와 서버간의 통신

3.2 ASP를 사용한 브라우저와 서버 사이의 통신

리스너가 성공적으로 서버에 연결된 후, 브라우저(클라이언트)는 그림 3과 같이 서버와 통신한다. 브라우저는 현재 실행중인 모든 리스너들의 목록을 얻어서 자신의 창에 표시한다. 브라우저는 서버의 ASP 페이지를 호출하는 방법으로 서버에 요청을 보낸다.

서버가 브라우저로부터 요청을 받으면, 현재 실행하고 있는 모든 리스너들의 목록을 데이터베이스에서 검색하

여 브라우저에게 XML 형태의 응답을 넘겨준다. 브라우저는 이 XML을 파싱한 결과를 사용자에게 보여준다.



▶▶ 그림 3. 브라우저와 서버간의 통신

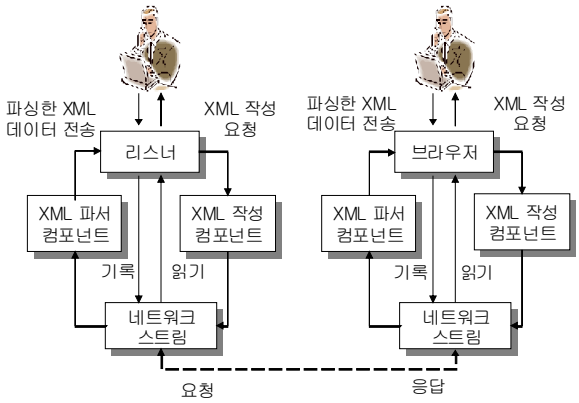
브라우저가 다시 서버의 도움을 받아야 할 때는 특정 파일에 대한 전역 검색을 수행할 때이다. 이 경우, 브라우저는 서버에게 특정 파일을 검색해 달라는 요청을 전송한다.

ASP 파일에 전달되는 매개변수들은 첫 번째는 사용자 이름에 대한 검색조건이고 두 번째는 공유된 파일에 대한 검색조건이다. 서버는 브라우저에게 넘겨받은 이 검색조건을 적용하여 데이터베이스에서 검색을 수행하고 그 결과를 XML 형태로 반환한다. 브라우저가 반환된 사용자 목록에서 특정 사용자를 선택하고 [열기] 버튼을 누르면, 브라우저는 선택된 리스너와 소켓연결을 수립한다. 즉 리스너와 브라우저 사이에 IP주소를 사용하여 직접 통신이 수립되는 것이다. 앞으로 수행될 브라우저와 리스너 사이의 모든 통신은 서버를 거치지 않고 이 소켓 연결 상에서 직접 이루어진다.

3.3 XML을 사용한 리스너와 브라우저 사이의 통신

이 어플리케이션의 통신 기능은 소켓에 기반을 두고 있는데, 소켓은 서로 통신을 하기 위해 시스템의 네트워크 스트림을 이용한다. 그림 4와 같이 리스너와 브라우저는 그들 각각의 네트워크 스티림에 데이터를 읽고 쓰는 작업을 수행한다. 요청/응답의 메커니즘이 이와 관련되어 지는데, 리스너는 네트워크 스트림으로부터 데이터를 읽어들이는 방법으로 요청을 처리한다.

응답 프로세스는 리스너가 브라우저의 네트워크 스트림에 데이터를 쓰는 방식으로 이루어진다. 브라우저는 리스너가 전송한 데이터를 얻기 위해 자신의 네트워크 스트림을 읽기 시작한다. 이때 전송되는 데이터는 XML 형식 또는 표준 바이트 포맷이다. XML 데이터가 양측에 도달하면, XML 파서는 수신된 XML 문서를 파싱하고 그 값을 반환해 준다. XML은 XML 작성 컴포넌트를 통해 생성된다. 만약 그 브라우저가 리스너에게 파일을 다운로드 혹은 업로드 하려고 하면, 브라우저는 XML 작성 컴포넌트를 통해 생성된 XML 요청을 전송한 후, 소켓 접속에서 파일을 직접 주고받는다. 리스너는 이 XML 데이터를 사전에 정의한 각각의 유형으로 분류하여 처리한다.



▶▶ 그림 4. 리스너와 브라우저간의 통신

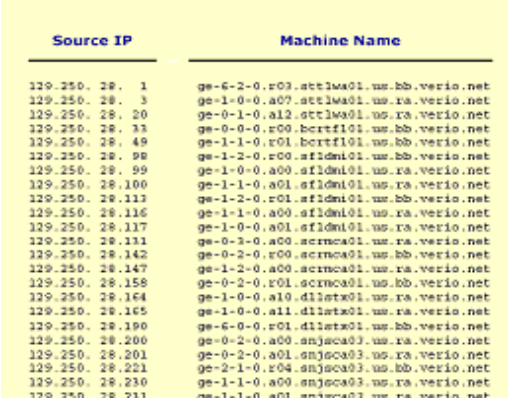
4. P2P에서의 보안 위협

4.1 DRDoS

차세대 DDoS 공격인 DRDoS(Distributed Reflection Denial of Service)는 2002년 1월 11일 오전 2시에 grc.com을 공격함으로써 처음 공개되었다. 이전 공격들은 반-스푸핑 UDP나 ICMP 플루딩, 또는 스푸핑 SYN 플루딩이 대역폭 소모 공격의 주류를 이루었지만, DRDoS는 SYN/ACK 플루딩을 이용한 공격이다. DRDoS 패킷이 다른 대역폭 소모 공격 패킷이 공격에 더 효과적인 방법을 제공하고 있지는 않는다. 로우소켓 프로그램을 이용해서 누구나 패킷을 자신이 원하는 대로 변경시킬 수 있기 때문이다[3][4].

가. grc.com에 대한 DRDoS 공격 사례

그림 5는 grc.com 서버에 대한 공격이 이루어지고 있을 때 기록된 로그 기록 일부이다.



Source IP	Machine Name
129.250. 28. 1	ge-6-2-0.r03.sttlwa01.us.bb.verio.net
129.250. 28. 3	ge-1-0-0.a07.sttlwa01.us.ra.verio.net
129.250. 28. 20	ge-0-1-0.a12.sttlwa01.us.ra.verio.net
129.250. 28. 33	ge-0-0-0.r00.bcrtfl01.us.bb.verio.net
129.250. 28. 49	ge-1-1-0.r01.bcrtfl01.us.bb.verio.net
129.250. 28. 98	ge-1-2-0.r00.sfldmi01.us.bb.verio.net
129.250. 28. 99	ge-1-0-0.a00.sfldmi01.us.ra.verio.net
129.250. 28.100	ge-1-1-0.a01.sfldmi01.us.ra.verio.net
129.250. 28.113	ge-1-2-0.r01.sfldmi01.us.bb.verio.net
129.250. 28.116	ge-1-1-0.a00.sfldmi01.us.ra.verio.net
129.250. 28.117	ge-1-0-0.a01.sfldmi01.us.ra.verio.net
129.250. 28.131	ge-0-3-0.a00.scrmca01.us.ra.verio.net
129.250. 28.142	ge-0-2-0.r00.scrmca01.us.bb.verio.net
129.250. 28.147	ge-1-2-0.a00.scrmca01.us.ra.verio.net
129.250. 28.158	ge-0-2-0.r01.scrmca01.us.bb.verio.net
129.250. 28.164	ge-1-0-0.a10.dllstx01.us.ra.verio.net
129.250. 28.165	ge-1-0-0.a11.dllstx01.us.ra.verio.net
129.250. 28.190	ge-6-0-0.r01.dllstx01.us.bb.verio.net
129.250. 28.200	ge-0-2-0.a00.snjsca03.us.ra.verio.net
129.250. 28.201	ge-0-2-0.a01.snjsca03.us.ra.verio.net
129.250. 28.221	ge-2-1-0.r04.snjsca03.us.bb.verio.net
129.250. 28.230	ge-1-1-0.a00.snjsca03.us.ra.verio.net

▶▶ 그림 5. DRDoS 공격에 대한 로그 기록

대부분이 Verio, Qwest, Above.net으로부터 들어온 패킷이며, 각각의 패킷을 살펴보면 소스 포트는 179번 TCP port이며, SYN와 ACK 플래그가 설정된 정상적인 패킷이다. 다시 말해서, grm 서버가 보낸 SYN 패킷에 대한 해당 ISP 라우터의 SYN/ACK 패킷이다.

179번 포트는 BGP(Border Gateway Protocol)가 사용하는 포트이며, 사용목적은 라우터가 패킷을 전달할 수 있는 IP 범위에 대해서 인접 라우터간에 라우팅 테이블을 교환을 위한 통신 시 이용된다. 세부적인 BGP 프로토콜은 중요하지 않다. 문제는 라우터가 179번 port를 통해서 들어오는 TCP Connection 요청을 승낙한다는 점이다. 다시 말해, 179번 포트를 통해서 들어오는 SYN 패킷에 대해서 해당 라우터가 SYN/ACK 패킷을 전송하게 된다.

grc 서버를 플루딩하기 위해서 이용된 라우터는 수 백여 개가 되며, 해당 라우터 모두가 해킹에 의해 제어 당했다고 볼 수 없다. 라우터에게서 발생된 모든 패킷은 SYN 패킷에 대한 정상적인 SYN/ACK 패킷이다. 따라서, 악의 있는 특정 호스트가 수 백개의 라우터를 대상으로 대상 호스트의 IP 주소로 스푸핑한 SYN 패킷을 179번 포트에 전송했고, 전달받은 라우터는 스푸핑된 IP주소를 가진 시스템에게 SYN/ACK 패킷을 전송하게 된 것이다. 반사 서버로 이용되는 라우터 목록을 확보하는 것은 그리 어려운 일은 아니며, trace route 명령어를 통해서 거쳐가는 라우트 정보를 확인하여 반사 서버로 이용할 수 있기 때문이다.

그러나, 이러한 공격을 막는 것은 어려운 일이 아니다. 직접적으로 라우터가 BGP 서비스 포트인 179번을 통해서 해당 웹서버에 패킷을 보낼 필요성은 없으며, 따라서 해당 포트로 들어오는 모든 패킷은 차단하면 라우터의 179번 포트를 통해서 들어오는 패킷 플루딩을 막을 수 있다. 그러나, 문제는 여기서 그치지 않는다. Grm 서버에 들어오는 179번 포트에 대한 패킷을 차단했지만, 또 다른 형태의 패킷 플루딩 공격이 이루어지고 있었다. 그림 6은 179번 포트에 대한 패킷 차단 후, 나타낸 공격에 대한 로그 기록 일부이다.

두 번째 SYN/ACK 플루딩 공격은 22번 포트(Secure Shell), 23번 포트(Telnet), 53번 포트(DNS), 그리고 80번 포트(HTTP/Web)를 이용하여 공격하고 있다. 일부 패킷은 4001번 포트(Proxy Server Port)나 6668번 포트(IRC 포트)를 이용하고 있다. 80포트를 이용한 반사 서버들의 특징은 매우 잘 알려진 Yahoo나 nsa 웹서버들을 사용하고 있다. 두 번째 플루딩 공격은 라우터의 179번 포트를 이용한 공격보다 반사 서버의 목록을 손쉽고, 더 많이 확보 할 수 있다.

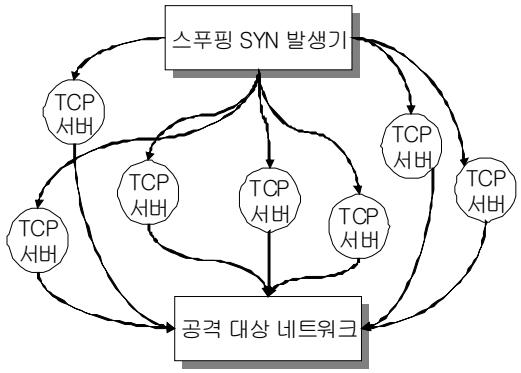
```
164.109. 18.251          whalenstoddard.com
171. 64. 14.238           www4.Stanford.EDU
205.205.134. 1        shell1.novalinktech.net
206.222.179.216            forsale.txic.net
208. 47.125. 33              gary7.nsa.gov
216. 34. 13.245     channelserver.namezero.com
216.111.239.132                www.jeah.net
216.115.102. 75          w3.snv.yahoo.com
216.115.102. 76          w4.snv.yahoo.com
216.115.102. 77          w5.snv.yahoo.com
216.115.102. 78          w6.snv.yahoo.com
216.115.102. 79          w7.snv.yahoo.com
216.115.102. 80          w8.snv.yahoo.com
216.115.102. 82         w10.snv.yahoo.com
```

▶▶ 그림 6. 다른 형태의 DRDoS공격에 대한 로그 기록

나. 패킷 경로 확산

공격 트래픽이 수많은 TCP 서버를 거쳐갈 때, 공격자는 패킷 경로 확산을 최대한 높이려고 시도 할 것이다. 인터넷은 각각 상호간에 연결된 거대 네트워크로 구성되어 있다. 인터넷에서 최종 사용자간에 주고받는 패킷들은 여러 요인에 의해서 거쳐가는 경로가 틀릴 수 있다. 또한, 인터넷 라우터가 거쳐가는 패킷에 대한 기록을 남기지 않기 때문에 역추적은 각각 거쳐온 라우터를 수동적으로 찾아가는 방법밖에 없다.

패킷 경로를 역추적 하는 것은 매우 어렵고, 수동적으로 역추적을 하는데는 많은 시간이 소비된다. 만약, 인터넷상에 수많은 Reflection Server가 존재한다면, 역추적은 더더욱 어려운 문제이다. 그림 7은 DRDoS 공격을 보여주고 있다.



▶▶ 그림 7. DRDoS 공격 구조

특정 공격자는 인터넷상에 넓게 흩어져 있는 반사 서버에게 스푸핑된 SYN 패킷을 보낸다. 이 패킷을 받은 반사 서버는 스푸핑된 목적호스트에 정상적인 연결 승인을 위해서 SYN/ACK 패킷을 전송하게 된다. 각각의 이들 패킷은 정상적인 패킷이며, 원하지 않는 무수히 많은 패킷을 동시에 받는 해당 호스트나 네트워크는 결국 마비되게 된다.

4.2 P2P 정보보호 대책

기관은 어느 시점에서 누구에 대해 어떤 정보를 공개하고 공유할 것인지, 또 PC 자원의 이용을 허가할 것인가에 대한 정보보호 정책을 명확히 하지 않으면 안 된다. 그래서 부서 내, 부서간이나 기관간, 사용자나 타 기관간 등에서 어느 P2P 소프트웨어를 이용하는가에 대해, P2P의 기술동향이나 타 기관의 도입 상황을 파악하고, 검사할 필요가 있다.

새로운 기술인 P2P라고 해도 그 기술은 기존의 TCP/IP로부터 시작되었으며 이것에 여러 가지 인터넷기술을 선택해서 구성하고, 거기에 새로운 기능을 추가하는 것으로 이루어진다. 따라서, 현재의 인터넷의 정보보호 대책을 P2P의 특성에 맞춰서 검토하는 것이 중요하다. P2P의 취약성에 대한 구체적인 정보보호 대책은 표 1과 같다[5].

표 1. P2P의 정보보호 서비스와 대책

기밀성	· 파일 접근 제어 · 파일의 암호화 · 접근 로그 관리 · 개인 방화벽 · 스팸메일 방지 · 전자 인증 시스템 · 파일의 기밀성에 따른 분류 · 사용자(기관내 · 외)의 제한 · 기관 내부 사용자의 신분 확인 · 사용자의 정보보호의무와 손해 배상 청구 계약의 체결
무결성	· 프로토콜에 대한 트래픽 제어 · 광대역 네트워크의 정비 · 소프트웨어의 재 전송 기능의 적용 · 데이터의 순차적 갱신 · 바이러스 대책 · 사용제한에 대한 규정
가용성	· 높은 가용성을 가진 부품의 사용 · 하드웨어의 이중화 · 데이터 백업 · 분산 병렬 처리 · IPv6의 적용 · 허가되지 않은 소프트웨어의 사용금지

5. 결론

P2P는 인터넷에서 중간에 서버 컴퓨터를 거치지 않고 정보를 찾는 사람과 정보를 가지고 있는 사람의 컴퓨터를 직접 연결시켜 데이터를 공유할 수 있게 해주는 기술과 그 기술을 응용해서 제공되는 서비스를 말한다. 인터넷에서 정보를 검색엔진을 거쳐 찾아야 하는 기존 방식과는 달리 인터넷에 연결된 모든 개인 컴퓨터로부터 직접 정보를 검색하고 제공받을 수 있다

그러나 P2P 서비스는 서버 없이 컴퓨터와 컴퓨터간에 데이터를 전송함으로서 의도적이거나 고의적인 공격자에 의해 보안 위협에 상당히 노출되어 있는 실정이다. 현재 보안업계에 따르면 국내에 본격 서비스되고 있는 P2P 방식으로 교환되는 파일에는 백오리피스 · 링제로 등 백도어 프로그램을 몰래 심어놓을 수 있어, 이를 알지 못하는 사용자는 자신의 컴퓨터를 쉽게 해킹 당할 수 있다. 더구나 공격자가 온라인 주식투자나 홈뱅킹에서 사용되는 패스워드를 알아낼 경우 경제적인 손실도 입게 돼 문제가 심각해지고 있다.

본 고에서는 P2P 어플리케이션의 구성과 P2P 어플리케이션의 구조들에 대해 기술하였다. 그리고 P2P에서 발생될 수 있는 공격중 DRDoS 공격 사례와 패킷 경로 확산에 대해 분석하였고 이러한 위협을 통해 안전한 P2P 서비스를 제공하기 위한 정보보호 대책을 살펴보았다.

참고문헌

[1] Buford, P2P Networking and Applications, Elsevier Science, 2008.

[2] Dreamtech Software Team, 조현석 편, P2P 애플리케이션 개발, 교학사, 2002

[3] 우훈식, JXTA: 차세대 P2P 플랫폼, 생능출판사, 2003
[4] DRDOS, htto://grc.com/dos/drdos.htm
[5] 김봉한, "P2P 환경에서 해킹 및 바이러스 대응 방안", 정보과학회지, 제22권 3호, 2004.

저 자 소 개

● 김 봉 한(Bong-Han Kim) 정회원

- 1994년 2월 : 청주대학교 전자계산학과(공학사)
- 1996년 2월 : 한남대학교 전자계산공학과 (공학석사)
- 2000년 2월 : 한남대학교 컴퓨터공학과(공학박사)
- 2001년 3월 ~ 현재 : 청주대학교 컴퓨터정보공학과 교수

<관심분야> : 멀티캐스트통신, P2P, 정보보호