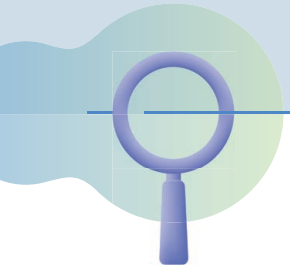

인터넷에서 사용되는 여러 기술

FTP 이야기 1

File
Transfer
Protocol

글
조인준
KBS 미디어기술연구부 수석연구원

네트워크의 가장 큰 목적은 연결된 장치들끼리 정보를 주고받는 것입니다. 우리가 회사나 가정에서 컴퓨터를 사용할 때도 알게 모르게 컴퓨터에 여러 정보가 오가고 있습니다. 이렇게 오고 가는 정보 중 많은 부분은 파일(File) 형태로 존재하는 것들입니다. 문서, 사진, 음악, 영상 등 우리가 다루는 대부분의 데이터가 파일이라는 단위로 저장되고 전송됩니다. 따라서 네트워크에서 파일을 어떻게 전송하는지가 매우 중요할 수밖에 없으며, 이를 위해 만들어진 대표적인 프로토콜이 바로 FTP(File Transfer Protocol)입니다. 여전히 편리하게 이용하고 있지만 익숙함 때문인지 자세히 들여다볼 생각을 하지 못한 경우가 많은 FTP에 대해서 작동 방식에 대한 대체적 이해가 가능한 수준까지 알아보도록 하겠습니다.

FTP에 대해 알아보기 전에 FTP의 전송 대상인 '파일'에 대해 간단히 알아보겠습니다. 파일은 컴퓨터에서 하나의 단위로 관리되는 정보의 묶음입니다. 예를 들어, 우리가 작성한 문서 등의 작업 결과물은 컴퓨터의 디렉터리나 폴더 안에 파일 형태로 저장됩니다. 파일은 보통 정보를 표현하는 문자나 기호 등이 이진 코드로 변환된 바이트들의 묶음이며, 읽기, 쓰기, 복사와 같은 작업도 이러한 파일 단위로 수행됩니다. 파일 자체에는 사용자가 저장한 실제 데이터가 담겨 있지만, 운영체제의 파일 시스템은 이와 별도로 파일 속성에 관한 추가 정보도 함께 관리합니다. 예를 들어, 파일의 크기, 생성된 날짜, 마지막 수정 시각 등은 파일 시스템이 관리하는 속성에 해당합니다.

컴퓨터에서 파일을 저장하는 방식은 여러 가지가 있는데, FAT32와 NTFS를 예로 들 수 있습니다. 이 둘은 파일의 내용은 똑같이 저장할 수 있지만, 파일에 붙여진 각종 추가 정보를 다루는 방법은 다릅니다. 예를 들어 NTFS에는 파일에 관한 권한 정보, 파일 소유자 정보, 압축이나 암호화 같은 복잡한 기능에 관한 정보 등이 있습니다. 그런데 FAT32에는 이런 기능이 없기 때문에 NTFS에서 FAT32로 옮기면 이런 정보들은 없어져 버립니다. 반대로 FAT32에서 NTFS로 옮길 때는, 원래 FAT32에는 없던 보안 정보 같은 것들이 NTFS에 새로 생기긴 하지만, 원래부터 있던 게 아니므로 그냥 기본 설정값으로 채워집니다. 즉, 파일 이름이나 크기, 만든 날짜 같은 기본적인 정보는 그대로 넘어가지만, 추가 정보는 일부 사라지거나 기본값으로 바뀔 수 있습니다.

파일은 그 자체로 온전한 정보 단위이기 때문에, 네트워크에서 파일을 옮기는 방식이 곧 정보 전송의 기본이 되었습니다. 실제로 파일 전송 프로토콜은 지금 우리가 쓰는 TCP/IP 같은 인터넷 프로토콜보다 먼저 등장하기도 했습니다. 즉, 파일 전송은 단순히 인터넷이 발전하면서 따라온 기능이 아니라, 정보를 담고 있는 파일 등을 안전하게 옮기기 위해 인터넷이 필요해진 측면도 있습니다. 오늘날 파일은 문서, 사진, 프로그램 등 그 어떤 정보라도 담을 수 있으며, 그 방법은 파일을 사용하는 사람이나 소프트웨어가

어떤 규칙에 따라 이진 바이트를 읽고 쓰느냐에 달려있습니다. 파일 전송 프로토콜이 인터넷 프로토콜보다 앞서 개발되었다고 말씀드렸듯이 실제로 FTP의 첫 번째 표준은 IETF(Internet Engineering Task Force)에서 1971년 4월에 발표한 RFC(Request for Comments) 114였습니다. 이때는 TCP/IP 프로토콜이 존재하기 전이었으며, RFC 114는 프로토콜의 기본 명령들과 이를 통해 장치들이 통신하는 방식을 정의했습니다. 당시에는 TCP/IP의 전신인 네트워크 제어 프로토콜(NCP, Network Control Protocol)이 전송에 사용되었습니다. 인터넷이 존재하지 않았고, 인터넷의 전신인 ARPAnet은 소수의 개발용 컴퓨터만 연결된 작은 네트워크에 불과했던 시절이었습니다.

그 후 여러 RFC를 통해 FTP 초기 버전이 다듬어졌습니다. 첫 번째 주요 개정은 1972년 7월 발표된 RFC 354로 현재 FTP의 가장 큰 특징인 제어 연결과 데이터 연결이 분리된 이중 연결구조가 처음 등장하였고, 지금도 사용하는 FTP의 많은 기능들이 기술되었습니다. 1973년 8월에 발표된 RFC 542는 오늘날 우리가 사용하는 FTP 사양과 유사했지만, 여전히 NCP 기반으로 정의되어 있었습니다. 이후 수많은 RFC를 통해 개정된 끝에, 큰 틀에서의 현대 FTP의 표준은 1980년 6월 발표된 RFC 765, File Transfer Protocol Specification에서 확립되었습니다. 이는 TCP/IP 상에서 FTP 동작을 정의한 최초의 표준으로, TCP/IP의 주요 표준들과 비슷한 시기에 만들어졌습니다. 1985년 10월에는 RFC 959, File Transfer Protocol(FTP)이 발표되었으며, RFC 765를 개정하면서 몇 가지 새로운 명령이 추가되었습니다. 오늘날 FTP의 기본 사양은 바로 이 RFC 959에 기반합니다. 이후에도 FTP 확장 기능, 보안 강화 기능 등을 정의하는 표준들이 발표되었습니다.

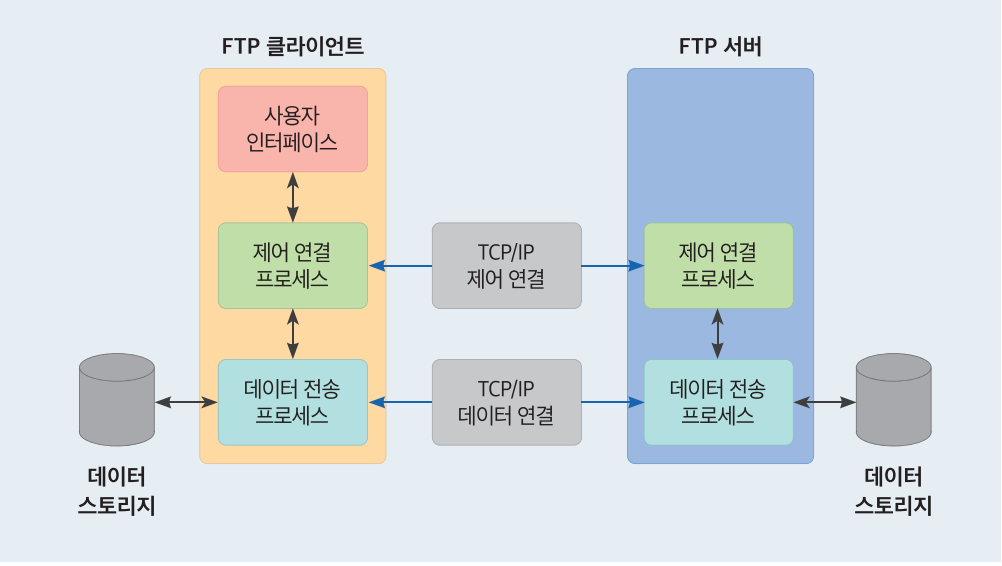

그림 1. FTP 클라이언트-서버 작동 구조

그럼 이제부터 [그림 1]을 통해 FTP의 동작 방식을 알아보겠습니다. [그림 1]에서 알 수 있듯이 FTP는 클라이언트/서버 프로토콜로써 클라이언트와 서버 간의 통신을 통해 이루어집니다. [그림 1]에서 한눈에 드러나는 특징 중 하나는 다른 많은 프로토콜처럼 TCP를 사용하지만 하나의 TCP/IP 연결만 사용하는 것이 아니라는 점입니다. FTP는 클라이언트와 서버 사이에 제어 연결(Control Connection)과 데이터 연결(Data Connection)이라는 두 개의 논리적 통신 채널을 기반으로 동작합니다.

☑ **제어 연결(Control Connection)**

FTP 세션(Session)이 설정될 때 만들어지는 TCP 논리 연결입니다. FTP 세션 동안 유지되며, 명령(command), 응답(reply), 상태 정보(status information) 등 제어 관련 정보만을 전송하는 데 사용됩니다. 실제 파일은 이 연결을 통해 전송되지 않습니다.

☑ **데이터 연결(Data Connection)**

서버에서 클라이언트로, 혹은 그 반대로 데이터를 전송할 때마다 별도의 TCP/IP 데이터 연결이 생성됩니다. 파일이나 데이터는 이 연결을 통해 전송되며, 전송이 완료되면 연결은 종료됩니다.

이처럼 제어와 데이터를 분리된 채널에서 서로 독립적으로 처리함으로써 여러 개의 파일을 차례대로 전송할 때 매번 로그인이나 세션 재설정이 필요 없고, 업로드와 다운로드 모두 동일한 세션 안에서 상황에 맞게 처리 가능하며, 서버와 클라이언트 중 데이터 연결을 어느 쪽이 시작할 것인지 선택할 수 있는 등의 프로토콜 유연성을 높일 수 있습니다.

FTP 서버 내부 프로세스는 서버 제어 연결 프로세스, 서버 데이터 전송 프로세스 두 가지 요소로 구성됩니다.

☑ **서버 제어 연결 프로세스**

서버 측 제어 연결을 관리하는 프로세스이며, 사전에 정의된 FTP 기본 포트(21번)에서 클라이언트로부터 들어오는 연결 요청을 수신 대기합니다. 연결이 설정되면, 사용자 프로세스로부터 명령을 받고, 이에 대한 응답을 전송하며, 서버의 데이터 전송 프로세스를 관리합니다.

☑ **서버 데이터 전송 프로세스**

서버 측의 데이터 전송을 관리하는 프로세스로, 클라이언트 데이터 전송 프로세스와 데이터를 송수신하는 역할을 담당합니다. 서버 데이터 전송 프로세스는 직접 데이터 연결을 설정하거나, 클라이언트가 시작한 데이터 연결을 수신 대기할 수 있습니다. 또한 서버의 로컬 파일 시스템과 상호작용하여 파일을 읽거나 기록하는 기능을 수행합니다.

FTP 클라이언트 내부 프로세스는 클라이언트 제어 연결 프로세스, 클라이언트 데이터 전송 프로세스, 사용자 인터페이스 세 가지 요소로 구성됩니다.

☑ **클라이언트 제어 연결 프로세스**

클라이언트 측의 제어 연결을 관리하는 프로세스이며, 서버 제어 연결 프로세스에 연결 요청을 보내 FTP 세션을 시작합니다. 연결이 설정되면 사용자 인터페이스에서 전달된 명령을 처리하여 서버 제어 연결 프로세스로 전송하고, 서버로부터 이에 대한 응답을 수신합니다. 또한 클라이언트의 데이터 전송 프로세스를 관리합니다.

☑ **클라이언트 데이터 전송 프로세스**

클라이언트 측의 데이터 전송을 관리하는 프로세스로, 서버 데이터 전송 프로세스와 데이터를 송수신합니다. 클라이언트 데이터 전송 프로세스는 직접 데이터 연결을 설정하거나, 서버에서 시작한 데

이터 연결에 대해 수신 대기할 수 있습니다. 또한 클라이언트 장치의 로컬 파일 시스템과 상호작용하여 파일을 읽거나 기록하는 기능을 수행합니다.

사용자 인터페이스(User Interface)

사용자 인터페이스는 쉽고 편리한 FTP 사용 환경을 제공합니다. 어렵게 느껴질 수 있는 FTP 명령 대신, 사람이 이해하기 쉬운 방식의 명령으로 FTP 기능을 수행할 수 있도록 하며, 세션을 운영하는 사용자에게 작업 처리 결과 및 관련 정보를 전달합니다.

FTP로 실제 파일을 전송하기 위해서는 데이터 연결을 수행하기 전에 반드시 제어 연결이 먼저 수립되어야 합니다. 제어 연결 수립 과정에서는 두 장치 간의 파일 전송에 사용할 FTP 세션을 생성하기 위해 정해진 절차를 수행합니다. 다른 종류의 클라이언트/서버 프로토콜들과 마찬가지로, FTP 서버는 제어 연결 과정에서 수동적으로 동작합니다. 서버 제어 연결 프로세스는 FTP 제어 연결을 위해 예약된 포트인 21번 포트에서 클라이언트의 연결 요청을 수신 대기합니다. 한편, 클라이언트 제어 연결 프로세스는 서버의 21번 포트에 TCP/IP로 제어 연결을 시작합니다. 이때 클라이언트가 TCP/IP 연결에 사용하는 포트는 임시 포트로 1024~65535 또는 49152~65535 범위에서 운영체제를 통해 자동으로 배정된 포트번호를 사용합니다. 클라이언트 제어 연결 프로세스에 의해 서버의 제어 연결 프로세스와 TCP/IP 연결이 수립되면, 클라이언트 제어 연결 프로세스가 서버 제어 연결 프로세스로 명령을 보내고, 이에 대한 응답이 서버로부터 클라이언트에게 보내집니다. 제어 연결이 정상적으로 동작하기 시작하면 가장 먼저 수행되는 절차는 사용자 인증 절차입니다.

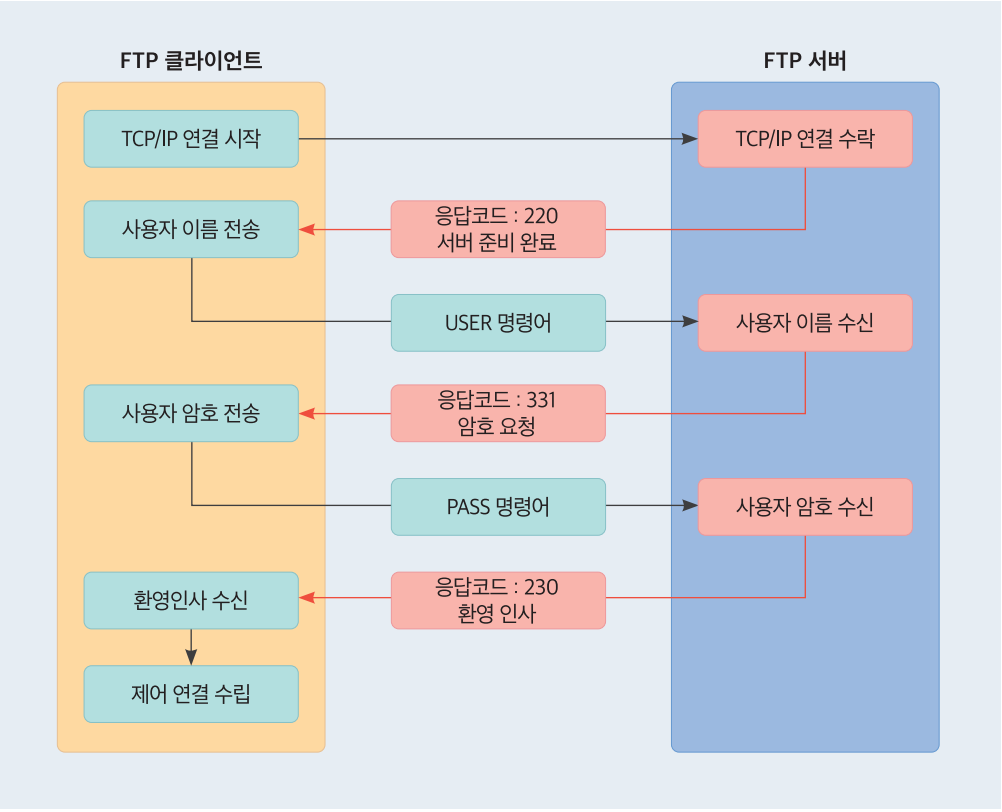

그림 2. FTP 사용자 인증 절차

FTP 로그인에서 사용되는 기본 인증 방식은 사용자 이름(Username)과 비밀번호(Password) 기반 방식입니다. 이는 인터넷을 포함한 다양한 환경에서 흔히 사용되는 인증 방식과 동일합니다. 클라이언트는 [그림 2]와 같이 TCP/IP 연결 후 USER 명령을 통해 클라이언트 제어 연결 프로세스에서 서버 제어 연결 프로세스로 사용자 이름을 전송합니다. 이어서 PASS 명령을 사용하여 비밀번호를 전송합니다. 서버는 전달받은 사용자 이름과 비밀번호를 내부 사용자 데이터베이스와 대조하여 접속하려는 사용자가 서버 접근 권한을 가졌는지 확인합니다. 정보가 올바르면 서버는 클라이언트에게 세션이 열렸음을 알리는 환영 인사(Greeting) 메시지를 보냅니다. 잘못된 사용자 이름이나 비밀번호를 입력한 경우, 서버는 인증을 다시 시도하도록 요청합니다. 여러 번 실패하면 서버는 타임아웃을 발생시키고 연결을 종료할 수도 있습니다.

인증이 성공적으로 이루어지면, 서버는 해당 사용자가 가진 권한 수준에 맞게 연결을 설정합니다. 예를 들어, 어떤 사용자는 특정 파일이나 정해진 종류의 파일에만 접근할 수 있으며, 어떤 사용자에게는 파일 읽기/쓰기 권한이 부여되는 반면 다른 사용자에게는 단순히 파일 다운로드만 허용될 수도 있습니다. 이처럼 관리자는 필요에 따라 FTP 접근 권한을 유연하게 설정할 수 있습니다. 연결이 설정되면, 서버는 사용자의 신원(Identity)에 따라 어떤 자원에 접근을 허용할지 결정을 내릴 수도 있습니다. 예를 들어, 다중 사용자가 존재하는 시스템에서는 관리자가 FTP를 설정하여 사용자가 접속할 때 자동으로 각자의 홈 디렉터리로 이동하도록 할 수 있습니다. 또한 사용자가 둘 이상의 계정을 가지고 있을 경우 특정 계정을 선택할 수도 있습니다.

FTP 사용자 인증 후 데이터 연결을 통한 전송 관련 내용으로 넘어가기 전에 보안에 관해 짚어보겠습니다. FTP는 인터넷 초창기부터 널리 사용되어 온 대표적인 파일 전송 프로토콜입니다. 그러나 FTP가 처음 설계된 시대는 오늘날과 같이 누구나 인터넷에 쉽게 접속할 수 있는 시대가 아닌 소수의 연구 기관과 대학 네트워크만 연결되어 있어 비교적 폐쇄적이고 신뢰가 높은 네트워크 환경의 시대였습니다. 이런 환경에서 FTP는 복잡한 보안 절차를 갖추지 않고도 운영될 수 있었으며, 사용자 이름과 비밀번호를 단순히 제어 연결을 통해 전달하는 가장 기본적인 형태의 로그인 절차만을 제공했습니다.
하지만 개방된 인터넷을 통해 정보를 전송해야 하는 현재의 기준으로 보면 이 방식은 매우 위험합니다. 제어 연결을 통해 주고받는 로그인 정보가 암호화되지 않은 상태로 그대로 노출될 수 있기 때문에, 네트워크 경로 중간에 있는 시스템이나 악의적인 공격자가 이를 손쉽게 가로챌 수 있습니다. 즉, 사용자 계정 정보가 그대로 노출되어 계정 탈취의 보안 위협에 놓이게 되는 것입니다.
이러한 한계를 극복하고자, IETF에서는 1997년 RFC 2228, FTP Security Extensions를 제정하여 FTP에 보안 확장 기능을 도입했습니다. 이 문서는 기존의 단순 로그인 방식을 보완하기 위해 보다 정교한 인증 메커니즘과 데이터 보호를 위한 암호화 옵션을 정의하고 있습니다. 이를 통해 FTP 클라이언트와 서버는 단순히 사용자 이름과 비밀번호를 주고받는 수준을 넘어, 보안 프로토콜을 연계하거나 암호화된 세션을 수립하여 안전하게 파일을 전송할 수 있게 되었습니다. FTP 보안 확장은 초기 신뢰 기반 네트워크에서 출발한 FTP가 오늘날의 인터넷 환경에서 생존하기 위한 필수 요소라고 할 수 있습니다. 보안 확장이 적용된 FTP는 기존 FTP의 편의성과 호환성을 유지하면서도 사용자 모두에게 보다 안전한 파일 전송 환경을 제공하고 있습니다.

그러나 아이러니하게도 모든 상황에서 보안이 절대적으로 필요한 것은 아니었습니다. 어떤 경우에는 오히려 복잡한 인증 절차 없이 누구나 접근할 수 있는 단순한 방식이 더 효율적일 때도 있습니다. 바로 이러한 배경에서 등장한 것이 익명 FTP(Anonymous FTP)입니다. 익명 FTP에 대해 처음 들으면 '왜 누구

나 FTP 서버에 마음대로 접속하도록 허용하는 걸까?'와 같은 의문 들기도 합니다. 하지만 그 이유는 생각보다 단순합니다. 익명 FTP는 불특정 다수의 일반 사용자에게 정보를 손쉽게 배포하기 위한 도구로 고안된 것입니다. 실제로 기업이나 기관이 대중에게 매뉴얼, 각종 안내문서, 소프트웨어 업데이트 파일 등을 제공해야 하는 경우, 모든 사용자에게 계정을 발급하고 비밀번호를 관리하는 것은 비효율적이고 번거로운 일이 됩니다. 오늘날 많은 기업이 이러한 배포 업무를 웹 서비스를 통해 처리하고 있지만, 웹이 보급되기 전인 1980년대~1990년대 초반까지는 FTP가 사실상 표준적인 배포 도구로 활용되었으며, 여전히 일부 기업이나 연구 기관 등에서는 FTP를 이용하고 있습니다.

이러한 공개 배포 목적에서는 모든 사용자에게 계정을 따로 발급하는 것이 효율적이지 않기에 1994년 RFC 1635, Anonymous FTP가 정의되었습니다. 이 방식에서는 클라이언트가 서버에 접속할 때 개별 사용자 이름 대신 anonymous 또는 ftp와 같은 기본 계정을 입력하여 접속합니다. 서버는 이를 확인한 뒤, 비밀번호 대신 이메일 주소를 입력하도록 요청하는 메시지를 돌려주기도 하는데, 이는 실제 보안을 위한 절차가 아니라 단지 접속자를 기록하기 위한 형식적인 데이터 수준에 불과합니다. 익명 FTP로 접속한 사용자는 게스트(Guest) 권한을 부여받아 사이트에 접근할 수 있지만, 그 권한은 매우 제한적입니다. 대부분은 읽기 권한만 부여되며, 지정된 공개 디렉터리에서 파일을 다운로드할 수 있을 뿐, 파일의 업로드, 삭제, 이름 변경 등의 행위는 허용되지 않습니다. 반대로 인증된 정식 사용자들은 전체 디렉터리 경로 탐색, 파일 관리(업로드, 삭제, 이름 변경 등)와 같은 훨씬 넓은 권한을 부여받을 수 있습니다. 결국, 익명 FTP는 보안을 낮추기보다는 파일을 누구나 쉽게 접근하고 받을 수 있도록 공개 자원을 배포하는 데 중점을 둔 특별한 방식입니다.

지금까지 FTP 프로토콜의 동작 모델과 제어 연결 및 이를 위한 인증 방식에 관해 설명해 드렸습니다.
다음 편에서는 데이터 전송 방식 등에 관해 설명해 드리겠습니다.

P.S.

C군이 여러분께 전하는 내용 중 전문적 성격이 짙은 것은 엄밀한 언어를 사용하여 설명하기에는 한계가 있습니다. 본 내용은 설명하는 대상에 대한 전체적 맥락의 이해에만 이용하시고, 그 이상은 권위 있는 전문자료를 참고하시기 바랍니다.