12. 기본 인증

서버는 사용자가 누구인지 식별할 수 있어야 한다.

- 사용자가 누구인지 알면, 어떤 작업이나 리소스에 접근할 수 있는지 결정할 수 있다.

12.1 인증

12.1.1 HTTP의 인증요구/응답 프레임워크

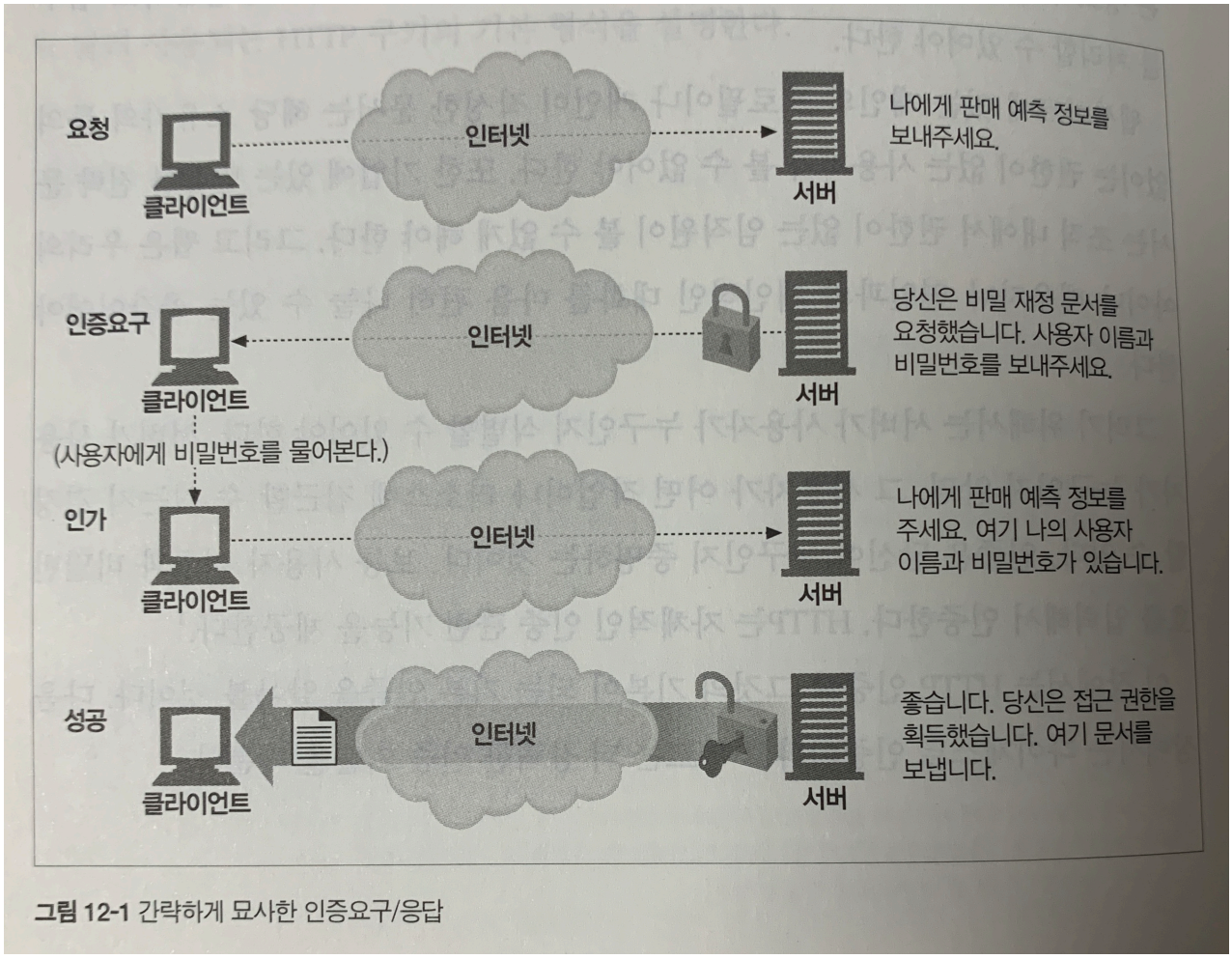
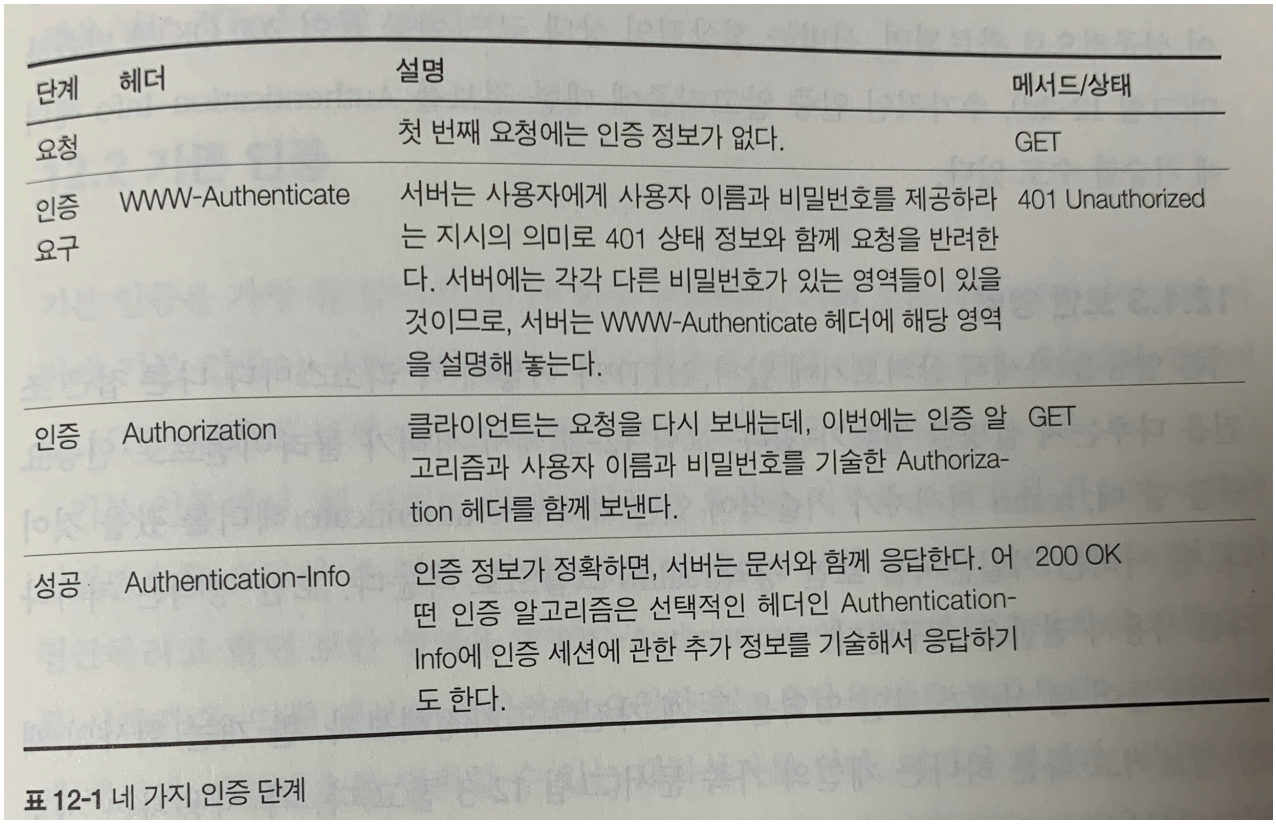


그림 12-1 간략하게 묘사한 인증요구/응답

12.2.2. 인증 프로토콜과 헤더

HTTP는 필요에 따라 고쳐 쓸 수 있는 제어 헤더를 통해, 다른 인증 프로토콜에 맞추어 확장할 수 있는 프레임워크를 제공한다.



단계	헤더	설명	메서드/상태
요청		첫 번째 요청에는 인증 정보가 없다.	GET
인증 요구	WWW-Authenticate	서버는 사용자에게 사용자 이름과 비밀번호를 제공하라는 지시의 의미로 401 상태 정보와 함께 요청을 반려한다. 서버에는 각각 다른 비밀번호가 있는 영역들이 있을 것이므로, 서버는 WWW-Authenticate 헤더에 해당 영역을 설명해 놓는다.	401 Unauthorized
인증	Authorization	클라이언트는 요청을 다시 보내는데, 이번에는 인증 알고리즘과 사용자 이름과 비밀번호를 기술한 Authorization 헤더를 함께 보낸다.	GET
성공	Authentication-Info	인증 정보가 정확하면, 서버는 문서와 함께 응답한다. 어떤 인증 알고리즘은 선택적인 헤더인 Authentication-Info에 인증 세션에 관한 추가 정보를 기술해서 응답하기도 한다.	200 OK

표 12-1 네 가지 인증 단계

12.1.3 보안 영역

HTTP는 각 리소스마다 다른 접근 조건을 다룰 수 있다. 웹 서버는 리소스를 `보안 영역(realm)` 그룹으로 나누며, 저마다 다른 사용자 권한을 요구한다.

```
HTTP/1.0 401 Unauthorized
WWW-Authenticate: Basic realm="Corporate Financials"
```

realm은 위와 같이 `해설 형식`으로 되어 있어서, 사용자가 권한의 범위를 이해하는 데 도움이 되어야 한다. 서버의 호스트명을 넣는 것도 유용할 수 있다.

12.2 기본 인증

가장 잘 알려진 HTTP 인증 규약으로 거의 모든 주요 클라이언트와 서버에 기본 인증이 구현되어 있다.

12.2.1 기본 인증의 예

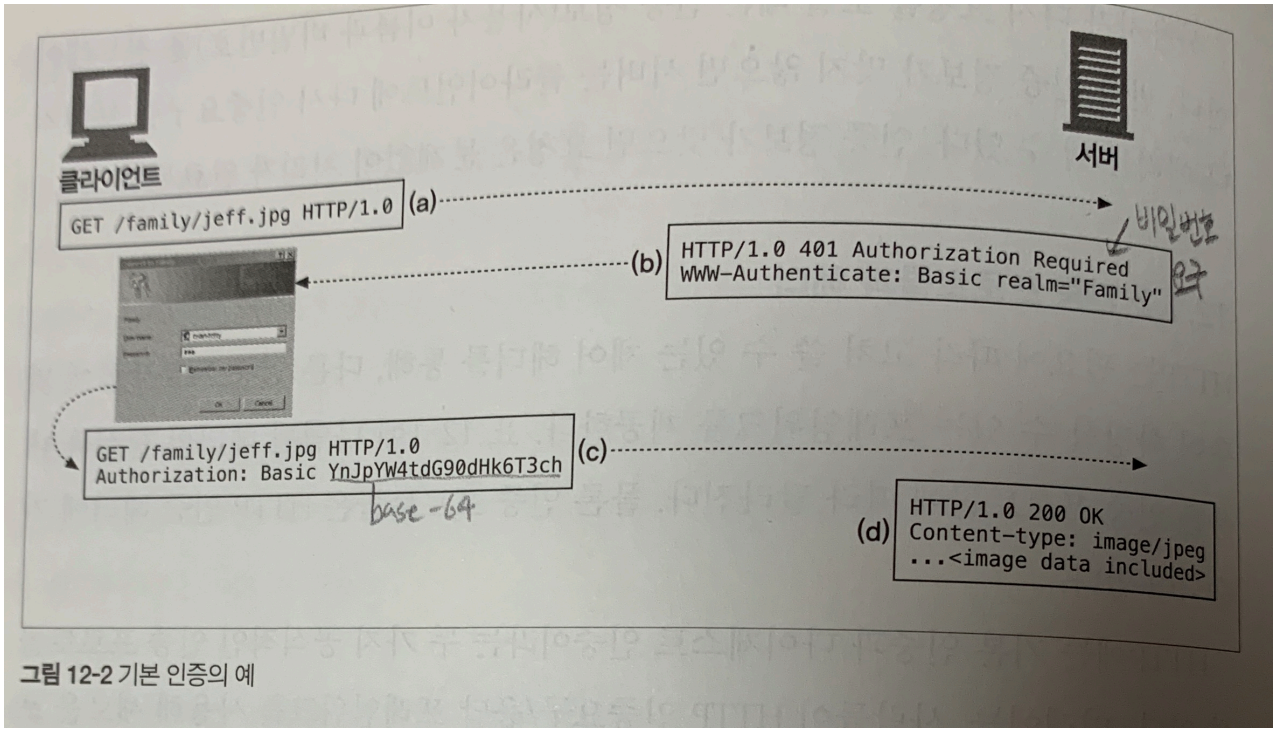


그림 12-2 기본 인증의 예

12.2.2 Base-64 사용자 이름/비밀번호 인코딩

HTTP 기본 인증은 사용자 이름과 비밀번호를 콜론으로 이어서 합치고, base-64 인코딩 메서드를 사용해 인코딩 한다.

base-64 인코딩: 8비트 바이트로 이루어져 있는 시퀀스를 6비트 덩어리의 시퀀스를 변환한다. 바이너리, 텍스트, 국제 문자 데이터 문자열을 받아서 전송할 수 있게, 그 문자열을 전송 가능한 문자인 알파벳으로 변환하기 위해 발명됐다. 전송 중에 원본 문자열이 변질될 걱정 없이 원격에서 디코딩할 수 있다.

12.2.3 프락시 인증

중개 프락시 서버를 통해 인증할 수도 있다.

프락시 인증은 웹 서버의 인증과 헤더와 상태 코드만 다르고 절차는 같다.

웹 서버	프락시 서버
비인증 상태 코드 : 401	비인증 상태 코드 : 407
WWW-Authenticate	Proxy-Authenticate
Authorization	Proxy-Authorization
Authentication-Info	Proxy-Authentication-Info

표 12-3 웹 서버 인증 vs 프락시 인증

12.3 기본 인증의 보안 결함

기본 인증은 일반적인 환경에서 개인화나 접근을 제어하는데 편리하며, 다른 사람들이 보지 않기를 원하기는 하지만, 보더라도 치명적이지 않은 경우에는 여전히 유효하다.

인코딩과 디코딩이 쉽다

base-64 방식으로 정보를 인코딩/디코딩 하는 경우 어렵지 않게 변환할 수 있다. 이게 문제가 된다면, 모든 HTTP 트랜잭션을 SSL 암호화 채널을 통해 보내거나, 보안이 더 강화된 다이제스트 인증 같은 프로토콜을 사용하는 것이 좋다.

재전송 공격

보안 비밀번호가 디코딩하기에 더 복잡한 방식으로 인코딩되어 있다고 하더라도, 여전히 제 3자가 중간에 가로챈 뒤 그대로 원 서버에 보내서 인증에 성공하고 서버에 접근할 수 있다. 기본 인증은 이러한 재전송 공격을 예방하기 위한 어떤 일도 하지 않는다.

동일한 사용자 이름과 비밀번호

사용자들은 여러 사이트에 동일한 사용자 이름과 비밀번호를 사용하기 마련이다. 때문에 보안이 중요하지 않은 애플리케이션이라고 하더라도 사용자의 정보가 노출되는 것은 위험하다.

기존 의도와 다른 요청

메시지의 인증 헤더를 수정하지는 않지만, 그 외 다른 부분을 수정해서 트랜잭션의 본래 의도를 바꿔버리는 프락시나 중개자가 중간에 개입하는 경우, 기본 인증은 정상적인 동작을 하지 않을 수 있다.

가짜 서버

기본 인증은 가짜 서버에 취약하다. 사용자는 가짜 서버나 게이트를 검증된 서버로 착각할 수 있다. 공격자는 사용자에게 비밀번호를 요청하고 그것을 나중에 사용할 목적으로 저장한 다음 에러가 난 척을 할 수 있다.