当请求需要授权认证的资源时，服务器会发起质询/认证过程。首次质询时浏览器会弹出一个会话，要求用户输入用户名和密码。之后浏览器会缓存该账号信息。并且在每次访问该资源时，在请求首部authorization中携带**账号信息**。

# 基本认证

基本上没有什么安全性可言

**服务器质询首部**：

* www-authentication：

1. BASIC/DIGEST：指定所用算法
2. Relm：密码作用域描述

**客户端授权首部**：

* Authorization：

1. BASIC/DIGEST：指定算法
2. Username/passwd：用户名和密码信息，BASE64编码

# 摘要认证

摘要认证有以下特点：

1. 使用摘要算法（例如MD5）对密码做摘要运算，最后在http报文中传递的是摘要。
2. 计算摘要时，输入有用户名、密码、服务器产生的随机数、报文头中的URL、method和实体内容等。因此，**能够检测出对报文首部和内容的篡改**。
3. 能够避免重放攻击。

## 摘要认证如何解决重放攻击

在计算摘要时，有一个输入是**对端产生的随机数**，随机数不同，那么产生的摘要也不同，从而达到避免重放攻击的目标，具体有以下几种产生随机数的方法：

1. **预授权**：即每次都在authentication-info中将**下一个随机数值**带给对端，这样保证了每一个http事务都是用到一个独一无二的随机数。这样哪怕攻击者截获了上一次http事务的密码摘要也没用，因为紧接着在下一次的http事务中由于使用了新的随机数，因此密码摘要也将发生变化。因此，预授权可以完全避免重放攻击。
2. **受限的随机数重用机制**：即给分配的每一个随机数指定一定的有效时间或有效使用次数。这样能够利用管线化进行优化传输，但是存在重放攻击的风险。
3. **同步生成随机数：需要两端共享密钥**