HTTP Authentication

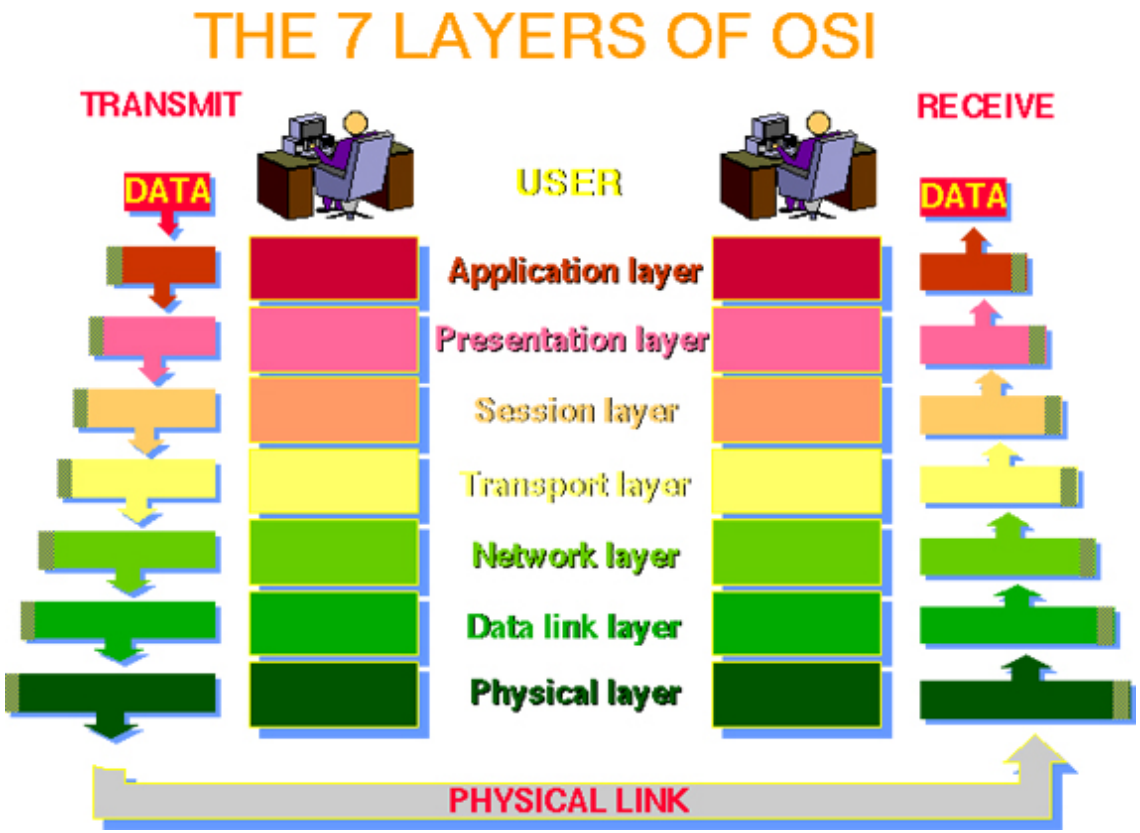
RFC 2617

obsoletes RFC 2069

Agenda

- Positioning
- Basic Access Authentication
- Digest Access Authentication
- Proxy-Authentication and Proxy-Authorization
- Security Considerations

Internet Layer



HTTP

Overview / Purpose

HTTP Authentication wants to provide a build in mechanism for requiering a valid username/password to gain access to web recourses

Overview

Initiated by:

- Webserver
- external cgi-script

Basic Access Authentication

Example: Client requests some Document in an protected Area

```
GET /download/report.doc HTTP/1.1
Accept: application/msword, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE
  5.01; Windows NT 5.0)
Host: 10.0.0.5:81
Connection: Keep-Alive
```

Basic Access Authentication

Example: Server reads its config-files

-> Server can only allow access to known users.

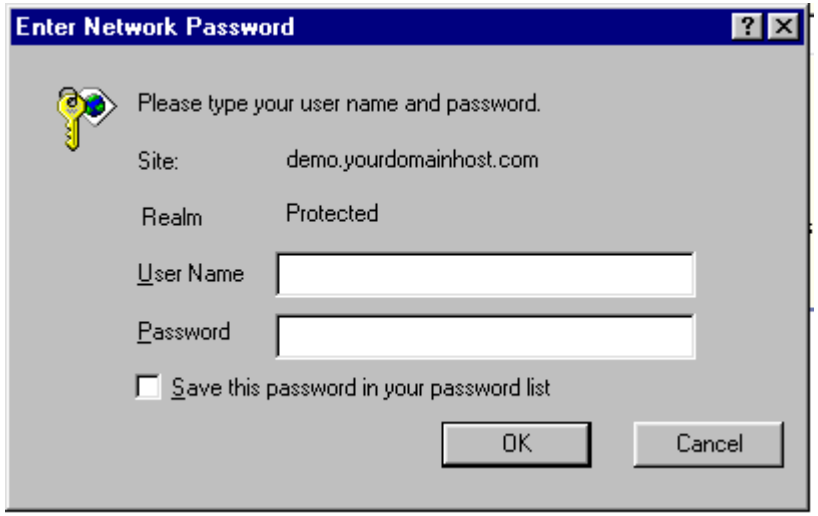
Basic Access Authentication

Example: Server Sends HTTP 401 Authorization Required Response

```
HTTP/1.1 401 Authorization Required
Date: Tue, 22 Jun 2004 03:54:06 GMT
Server: Apache/1.3.29 (Unix)
WWW-Authenticate: Basic realm="Protected"
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
```

Basic Access Authentication

Example: Browser displays Username/Password prompt displaying host name and authentication realm.



Basic Access Authentication

Example: Client Resubmits Request with Username/Password

```
GET /download/report.doc HTTP/1.1
Accept: application/msword, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE
  5.01; Windows NT 5.0)
Host: 10.0.0.5:81
Connection: Keep-Alive
Authorization: Basic ZnJhbms6ZmllZGxlcg==
```

Basic Access Authentication

Example: Server compares client information to its user/password list.

3 Possibilitys:

- username : password is valid:
  server sends requested content.
- authorization fails:
  server resends 401 Authorization Required header
- Client hits cancel:
  browser shows error message sent along with 401 message.

Basic Access Authentication

Problem: The Username/Password is sent in cleartext

Authorization: Basic ZnJhbms6ZmllZGxlcg==

```
"ZnJhbms6ZmllZGxlcg==" ->
  base64decode() ->
"frank:fiedler"
```

Basic Access Authentication

Solution:

Digest Access Authentication

- Password won't be sent in cleartext
- Password will be sent encrypted (normaly as md5 hash of the password and some other values)

Digest Access Authentication

Additional required Headers:

Server requests Authorisation:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
  realm="Protected",
  qop="auth,auth-int",
  nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
  opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

Digest Access Authentication

Description of the aditional attributes

realm: Displayed to User in Login-Formula
qop: quality of protection
for backward compatibility with RFC 2069
- The value "auth" indicates authentication;
- the value "auth-int" indicates authentication with integrity protection
nonce: server-specified quoted data string uniquely generated each time a 401 response is made.
opaque: quoted data string replied unchanged the whole session by the client; might be used for session tracking
stale: flag set if the client requests a new nonce value
TRUE: - if the client wants to reauthenticate
- if the server gets an outdated nonce value but correct user/password from the client
algorithm: one or more algorithms used to encrypt user/password

Digest Access Authentication

Additional required Headers:

Client replies Authorisation:

```
Authorization: Digest username="frank",
               realm="Protected",
               nonce="dcd98b7102dd2f0e8b11d0f600bf
                      b0c093",
               uri="/dir/index.html",
               qop=auth,
               nc=00000001,
               cnonce="0a4f113b",
               response="6629fae49393a05397450978507c
                         4ef1",
               opaque="5ccc069c403ebaf9f0171e9517f
                       40e41"
```

Digest Access Authentication

Description of the aditional attributes

username: the username in cleartext
realm: the realm the user wants to authenticate to
qop: the quality of protection selected by the client
- must be present if the server sent a qop directive

cnonce: client generated unique data string
- must be present if qop is present

nc: nonce-count - the count of requests sent by the client
- must be present if qop is present

response: encrypted password

Digest Access Authentication

How the response is encrypted

depending on qop its the md5 Hash of various attributes

Proxy-Authentication and Proxy-Authorization

This authentication scheme may also be used for authenticating users to proxies, proxies to proxies, or proxies to origin servers by use of the Proxy-Authenticate and Proxy-Authorization headers.

Proxy-Authentication and Proxy-Authorization

Just replace the Authentication-Header:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
...
```

would be:

```
HTTP/1.1 407 Proxy Authentication Required
  Proxy-Authenticate: Digest
  ...
```

Security Considerations

Basic Authentication

Very insecure because of cleartext transmition of username/password (Man in the middle/Network sniffering)

Security Considerations

Digest Authentication

- Replay Attacks

Depending on the way the nonce-value is created

can be completely avoid (if nessecary) if each nonce-value is only used once

Security Considerations

Digest Authentication

- Multiple Authentication Schemes

some Browsers only support Basic Authentication

Security Considerations

Digest Authentication

- Online dictionary attacks

to avoid them force the usage of "strong" passwords, not listed in any dictionary

Security Considerations

Digest Authentication

- Man in the Middle

- remove all offered choices, replacing them with a challenge that requests only Basic authentication (may realized as http-proxy)

-> Useragents should display the auth-mechanism

Security Considerations

Digest Authentication

- Man in the Middle

- eve sends the same nonce to more clients -> time to find the first pwd will be reduced
  -> if one password is known all passwords can be decrypted

can be avoid using the cnonce-directive by the clients

Security Considerations

Basic/Digest Authentication

Password-File at the Server

- stored at a safe location!
- passwords stored not as cleartext

Conclusion