An Extension of the HTTP Authentication Scheme To Support Server Groups

1. Status of this Memo

2. Abstract

This document motivates and describes an extension to HTTP which allows protection spaces to be extended across multiple servers residing in possibly different domains. These servers form groups that allow browsers to obtain authentication information from a user just once while accessing information on any one server cooperating in such a group.

To achieve this behavior, the HTTP WWW-Authenticate header information must be extended. This approach is independent of the authentication scheme, but is most scalable in conjunction with a

trusted third party authentication scheme, such as the proposed Mediated Digest Authentication.

3. Acknowledgments

Thanks to Guenther Karjoth and Markus Buehler who helped identify the need for this proposal.

Contents

4. Introduction

In this paper we propose an addition to the HTTP authentication information which allows users to provide their authentication information once for an entire set of servers belonging to the same logical group. The concept is that even if servers reside in different Internet domains, this enhancement allows them to share information among authorized members in a seamless manner.

Although the proposed extension is not restricted to any particular authentication technique, we base our work on the Mediated Digest Authentication [4] Internet-Draft proposal, whereby authentication information is protected and, in the context of cooperating servers, group management is facilitated.

5. HTTP Authentication and Protection Space Model

Authentication of HTTP V1.0 [1] uses a simple challenge-response mechanism, which works as follows: a server which requires clients to authenticate themselves replies to a client's HTTP method invocation with an Unauthorized (401) error code, and an authentication challenge that contains an indication of the required authentication method. The client may then resubmit its request, but must include authentication information with it to be allowed access to the server's resources. The authentication challenge is always tied to a server-defined protection space.

The protection space is defined through a combination of the server's root Uniform Resource Locator (URL) and a server-specified realm. For all practical purposes, the root URL is the server's name in either DNS or dotted IP address notation. The realm is a name the server administrator chooses and can be used to partition the server's file system into different protection spaces.

In a typical implementation, encountering an Unauthorized (401) response causes a client browser to prompt the user for authentication credentials before the request is resubmitted with this information. For future accesses to the same protection space, the browser may then cache the user credentials and automatically include them in its accesses without further prompting the user.

HTTP V1.0 [1] specifies only a basic authentication mechanism (BA), where the browser includes the user identification and password as a base-64 encoded string in an authentication field in the request header. This scheme is described in the draft as "a non-secure method of filtering unauthorized access to resources on an HTTP server". As this authentication information travels in the clear over an essentially insecure network, it is susceptible to capture by an observer who could then masquerade as the original user.

To address the security concerns of the basic HTTP authentication method, two proposals were made in 1995. Both ensure that the authentication information never travels in the clear. These schemes can also prevent replay of HTTP requests by requiring that part of the request information changes on subsequent accesses by a same user to a given server protection space. This is achieved via the inclusion of nonces.

Digest Authentication (DA) [2] operates in a similar fashion to BA, but transfers the user-id and password from browser to server in a hashed form. A server provided nonce can be used to establish response freshness and detect replays. Mediated Digest Authentication (MDA) [4] is based on DA, but introduces trusted third-parties who act as mediators in proving identity and establishing a session key. Introduction of a client nonce enables the browser to distinguish the server reply as the valid response associated with its request. A server employing MDA supplies a list of trusted third-parties, with which the server shares a cryptographic key, to a requesting client. If the client is able to identify a third-party in this list with which it also shares a cryptographic key, then this party can be contacted to enable mutual authentication to occur.

6. The Server Group Model And Assumptions

In this proposal we define a group of servers as a set of physically distinct servers wishing to collaborate to provide information to a closed user group. This closed user group is trusted by each server.

A server group can be recognized as existing amongst the involved servers where, from a security point of view, access to one 'group member' ensures access to other group members. In practical terms this means that a company or organization can distribute information over several (geographically- and/or domain-separated) servers to form a server group.

In the case of BA or DA, this requires that the (same) user-id and password are stored at each server and kept consistent among these.

For reasons of consistency and distribution, use of the MDA scheme, which employs trusted third party based authentication, is very attractive. By 'signing-on' to any one of these group servers, a user can access any other server of that server group without having to repeat the sign-on procedure.

7. Secure Authentication To A Group Of WWW Servers

To achieve the server group model described, single server authentication requires extension to allow seamless retrieval of documents from any server belonging to a group. Secure access to servers in the group is seamless in that user-id and password are explicitly supplied by the user only once (exploiting the normal caching of authentication information in the browser), for the first group server encountered.

As described in section 5, HTTP authentication information is always relative to a server's protection space, which is based on the combination of the canonical root URL of that server and a server defined realm. As a root URL always gives the server's Internet address, the protection space is uniquely tied to that server and cannot be extended to other servers that are only known under different Internet addresses.

To extend the HTTP protection space beyond a single WWW server, the definition of the protection space must be made relative to a unique group name rather than a (root) URL, and the browser must be notified that the protection space information pertains to a group of servers rather than a single server. In subsequent sections we describe how this can be achieved.

8. Group Names and HTTP Modification

We considered several different solutions for defining unique group names for the global WWW. X.500-like distinguished group names were one candidate, but we decided that (for the time being) the overhead associated with maintaining such a naming scheme would be excessive. We rather elected to use the DNS name of a distinguished server in the server group as a unique base for the group name. This server name combined with a freely chosen group name, and the realm of the original HTTP specification, then defines a protection space that can span multiple servers belonging to the same server group.

The required modifications to HTTP are described in the next two subsections.

8.1. WWW-Authenticate Response-Header

The WWW-Authenticate response header is prescribed for Unauthorized (401) server response messages. The field value is defined as consisting of at least one challenge that indicates the authentication scheme(s) and parameters applicable to the Request-URI.

The challenge-response authentication mechanism of HTTP is described in [1] as:

```
auth-scheme = token

auth-param = token "=" quoted-string
```

where 'token' is a case-insensitive identification of the authentication scheme, or the attribute for which an associated value is given, respectively.

The 401 (unauthorized) response is structured as follows:

```
challenge = auth-scheme 1*SP realm *("," auth-param)


realm = "realm" "=" realm-value
realm-value = quoted-string
```

The HTTP V1.0 draft limits the 'realm' to a specific server where 'the realm value, in combination with the canonical root URL of the server being accessed, defines the protection space'. In this way a set of 'protection spaces' is achieved. It is explicitly stated, though, that a single protection space cannot extend outside the scope of its server unless the authentication scheme specifies otherwise.

Our proposal is to add an optional 'servgrp' field which can be used independently of authentication scheme to achieve cross-server protection spaces when desired. This results in a challenge field consisting of:

```
challenge = auth-scheme 1*SP realm *("," auth-param) ["," servgrp]

servgrp = "servgrp" "=" servgrp-value

servgrp-value = quoted-string
```

The content of the servgrp-value should be made unique in accordance with the naming proposals in section 8 of this document.

8.2. Authorization Request-Header

The Authorization request-header field is included by a user agent wishing to authenticate itself with a server. The field value is described in [1] as consisting of credentials containing the authentication information of the user agent for the realm of the resource being requested. The specification of a realm by a server indicates that the same credentials should be valid for all other requests within the nominated protection space.

9. Example Usage

In this section we illustrate the usage of the servgrp field with the Basic[1], Digest[2] and Mediated[4] authentication schemes.

We have arbitrarily chosen to use the server "hundshorn.zurich.ibm.com" as the unique group identifier, and illustrate three authentication schemes with a group:

hundshorn.zurich.ibm.com:med_image_group

9.1. Basic Authentication

The following message shows the WWW-Authenticate header containing server group information:

```
WWW-Authenticate:  Basic realm="x-rays",
servgrp="hundshorn.zurich.ibm.com:med_image_group"
```

The browser constructs an unaltered BA Authorization response with base-64 encoded user-id and password:

```
Authorization:  Basic YWh1OmFodQ==
```

9.2. Digest Authentication

Extension of the digest authentication scheme results in a similar extension of the WWW-Authenticate header with the addition of the server group information:

```
WWW-Authenticate:  Digest realm="x-rays", nonce="824385109",
opaque="ef609f5de336fc7f0e0f5e22da9c921c",
servgrp="hundshorn.zurich.ibm.com:med_image_group"
```

In this scheme the browser also returns an unaltered DA Authorization response. In this case the browser returns a digest of the user-id and password, so that an observer could not directly retrieve these fields as in the BA case:

```
Authorization:  Digest username="ahu", realm="x-rays",
nonce="824385109", uri="digest/",
response="b9686783c8e29f91417a557bd13f65b7",
opaque="ef609f5de336fc7f0e0f5e22da9c921c"
```

9.3. Mediated Digest Authentication

Extension of the MDA scheme can also be seen to present additional server group information with the WWW-Authenticate header fields. The MDA proposal also adds Trusted-Party messages to the server's reply. A mutually trusted party is contacted by the browser to perform authentication. The information in the server-mac field is passed by the browser to the selected authentication server (trusted-party), and in this way the client is also able to authenticate the server. This is not possible in BA or DA:

```
WWW-Authenticate:  Mediated realm="x-rays", nonce="824384997",
opaque="017a975b3e5972507e7e098dc23e2031",
servgrp="hundshorn.zurich.ibm.com:med_image_group"

Trusted-Party:  uri="mdap://gletscherhorn.zurich.ibm.com:1995",
server-name="hundshorn.zurich.ibm.com",
server-mac="fb400c1e1dc023e49441cd5092fdd566"

Trusted-Party:  uri="mdap://galmihorn.zurich.ibm.com:1995",
server-name="hundshorn.zurich.ibm.com",
server-mac="e0dde363778ee29540b4410ab009daac"
```

```
Trusted-Party: uri="mdap://hundshorn.zurich.ibm.com:1995",
server-name="hundshorn.zurich.ibm.com",
server-mac="0205a4ea82fa6b596025038326759298"
```

After receiving a (positive) response from the contacted trusted-party, the browser returns Authorization and Session-key headers to the server:

```
Authorization: Mediated username="ahu", realm="x-rays",
nonce="824384997",
uri="mediated/", response="12f1950f5ca91f4febb7c6c05d0ec284",
opaque="017a975b3e5972507e7e098dc23e2031"

Session-key: uri="mdap://hundshorn.zurich.ibm.com:1995",
server-name="hundshorn.zurich.ibm.com",
server-key="ce6b7e04611d687ebdae5bdb59514593",
server-mac="0dcda62dc189d186302b81f97c275827"
```

The Internet Draft on MDA [4] requires a WWW server to cache session keys established by the authentication server. In this way state is introduced into the server which is supposed to be stateless. We therefore decided to cache the session key data received from the authentication server in the user agent. The user agent then resends this information each time a particular server is accessed again. Alternatively the approach of [3] could be deployed by adding the session key to the State-Info header (in encrypted form).

10. Implementation Issues

In this section we briefly describe the issues which have to be addressed in supporting the HTTP Modification described in section 8.

10.1. Server

On the server side it is necessary to enhance the access control files with a directive for naming the server group to which documents belong.

10.2. User Agent

On the user agent side it is necessary to take the group organization into consideration when caching authentication information. The current organization which HTTP implies (via the uni-server 'realm') is that servers are divided into local protection spaces. In the proposed multi-server 'server group' protection space, cache lookup should first consider group membership before attempting realm matching.

11. Conclusion

This document has presented a mechanism by which the existing HTTP single-server protection space can be expanded to provide protection across multiple servers. The extension enables seamless cooperation amongst servers, and secure authentication if it is used with a method such as MDA.

The nature of this proposal is such that the enhancements can be implemented to coexist with the authentication mechanism of HTTP V1.0, or the various proposed authentication enhancements. The proposal allows upward compatibility, in that non group-enabled browsers simply ignore the additional information.

In the long term it is also proposed that secure group enablement should be incorporated into HTTP V1.1 as a permanent feature of the revised protocol.

References

[1] T. Berners-Lee and R. T. Fielding and H. Frystyk Nielsen HyperText Transfer Protocol (HTTP) Internet Draft, Work in Progress February 1996

[2] J. Hostetler and others A Proposed Extension to HTTP: Digest Access Authentication Internet Draft, Work in Progress December 1995

[3] D. M. Kristol and L. Montulli Proposed HTTP State Management Mechanism Internet Draft, Work in Progress February 1996

[4] D. Raggett Mediated Digest Authentication Internet Draft, Work in Progress March 1995 (expired)

12. Authors' Address

Andrew Hutchison, Matthias Kaiserswerth, Peter Trommler
IBM Zurich Research Laboratory
Saeumerstrasse 4
CH-8803, Rueschlikon
Switzerland
{ahu,kai,trp}@zurich.ibm.com

Any correspondence should please be directed to trp@zurich.ibm.com.
(Phone: +41 1 724 8373 Fax: +41 1 710 3608)