Using UsernameToken Authentication with Blogger and MetaWebLog APIs

Vadim Zaliva, lord@crocodile.org

February 19, 2007

1 Background

Blogger[2] API is a popular API used by many online services for blog posting and editing. MetaWebLog [1] is extension of Blogger API which allows to get and set attributed of blog posts. Particularly, it allows to append media files to blog posts. Both Blogger and MetaWebLog APIs are using XML-RPC[3], a simple XML-based RPC protocol used over HTTP. Both Blogger and MetaWebLog APIs are using clear-text username/password authentication.

2 UserToken Authentication

Blogger or MetaWebLog API endpoint, could optionally provide added security by allowing user to use alternative authentication mechanism. The proposed extension is based on HTTP authentication mechanism[5] using Username Token[4] and similar to one, used in ATOM API[6]. UsernameToken authentication scheme is resilient to replay attacks and does not allow to recover user password by recording client/server protocol exchange.

The main idea of UsernameToken authentication profile is that a client calculates a *Password Digest* from following three pieces of information:

1. One-time token (also called "Nonce")
2. Timestamp (when request is generated)
3. Password

They are concatenated, and their hash is calcumated using using SHA1[10] Secure Hash Algorihm. The hash is encoded using BASE64 encoding. Each HTTP request sent by client contains "X-WSSE" HTTP header with the following fields:

Username - User id

PasswordDigest - Password Digest calculated as described

Nonce - One-time token, used during digest calculation

Created - Timestamp, used during digest calculation

For informal descirption see [9]. For more formal specification please refer to [4].

3 Compatibility

For backwards compatibility, the signatures of all Blogger and MetaWebLog API methods remain the same. They still expect *username* and *password* parameters. However, if client has provided alternative authentication information per this document, the *username* and *password* parameters in API calls are ignored and authentication is performed based on information provided via HTTP protocol headers. A user name provided in "X-WSSE" header supersedes the value *username* parameter in API calls.

A server could support both new and old authentication models. If server supports only new model (deeming old one not sufficiently secure), it should reject all non-authenticated requests with 401 HTTP error code. For example, server response:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: WSSE realm="foo", profile="UsernameToken"
```

Indicates that client has not provided satisfactory credentials using "WSSE" authentication scheme using "UsernameToken" authentication profile for realm "foo".

Authentication failures caused by username/password mismatch when old authentication scheme is used are reported by return XML-RPC *Fault* response from API method.

4 Examples

Client request to Blogger API "getUsersBlogs" method using UsernameToken authentication:

```
POST /API/V1 HTTP/1.1
Host: blog.example.com
Content-Type: text/xml
Authorization: WSSE profile="UsernameToken"
X-WSSE: UsernameToken Username="lord", PasswordDigest="quR/EWLAV4xLf9Zqyw4pDmfV9OY=", Nonce=
Content-length: 515

<?xml version="1.0"?>
<methodCall>
  <methodName>blogger.getUsersBlogs</methodName>
```

```
  <params>
    <param><value><string>1234ABCDER</string></value></param>
    <param><value><string></string></value></param>
    <param><value><string></string></value></param>
  </params>
</methodCall>
```

References

[1] Dave Winer: "MetaWebLog API"

[2] Dave Winer: "Blogger API"

[3] Dave Winer: "XML-RPC Specification"

[4] "Web Services Security UsernameToken Profile 1.0"

[5] "RFC 2617 - HTTP Authentication: Basic and Digest Access Authentication"

[6] "The Atom API"

[7] "LifeBlog Posting Protocol Specification"

[8] "PodShow Authoring API"

[9] Mark Pilgrim: "Atom Authentication"

[10] RFC 3174 - US Secure Hash Algorithm 1 (SHA1)