Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: March 10, 2013

E. Prodromou
StatusNet, Inc.

September 6, 2012

HTTP Authentication: Dialback Access Authentication
draft-prodromou-dialback-00

Abstract

This specification defines the Dialback Access Authentication Scheme. It provides a way for HTTP clients to identify an Internet host or account responsible for an HTTP request, and for HTTP servers to verify that identity by sending a token to a declared dialback endpoint.

The specification defines a new HTTP authentication scheme, "Dialback". It also defines a new link relation, "dialback", to specify the endpoint for the dialback verification. Finally, it defines the interface for the dialback endpoint.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 10, 2013.

Copyright Notice

Table of Contents

1. Introduction

HTTP/1.1 [RFC2616] has an extensible authentication mechanism using the "Authorization" header. Basic and Digest Authentication [RFC2617] are two authentication schemes for HTTP/1.1. OAuth 1.0 [RFC5849] is another.

All of these authentication schemes assume that the HTTP server is able to validate the authentication credentials. With distributed publishing and social networking applications, however, the security domain may be separate from the resource domain. In addition, the resource and security domains may have no previously-negotiated relationship.

With dialback authentication, an HTTP client can authenticate as a host or Webfinger address without creating a previous relationship. The HTTP server verifies the identity using a dialback endpoint specified by the host or Webfinger address.

Because dialback authentication requires one or more additional requests from server to client, its intended use is for bootstrapping longer-term relationships, such as dynamic registration of OAuth clients. It can also be useful for single use requests.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Authentication scheme

This spec adds a new Authorization type, "Dialback".

The header has the following optional elements:

host The host authorizing the request.

webfinger The webfinger account authorizing the request.

token An opaque value used to confirm the request authorization

Exactly one of "host" or "webfinger" is required.

"token" is always required.

3. Dialback endpoint discovery

To validate the token, the server must identify a dialback endpoint for the host or Webfinger address.

For a host parameter, the server SHOULD use Web Host Metadata [RFC6415] to find the endpoint. It will have the link relation "dialback".

For a webfinger parameter, the server SHOULD use Webfinger [I-D.jones-appsawg-webfinger] to find the endpoint with the link relation "dialback".

4. Dialback verification endpoint

To confirm, the server makes an HTTP POST request to the dialback endpoint. The request MUST have the Content-Type application/x-www-url-encoded.

The HTTP request has the following parameters.

host The host value provided in the original Authorization header.

webfinger The webfinger value provided in the original Authorization header.

token The token provided in the original Authorization header.

url The URL that the original request was made to.

date The Date header on the original request.

The request MUST include exactly one of "host" or "webfinger".

The request MUST include the "token", "url" and "date" parameters.

All parameters MUST be encoded in the body of the request.

If the token is valid, the endpoint SHOULD return a 200 OK or 204 No Content result.

If the token is invalid, the endpoint SHOULD return a 400 Bad Request result.

5. Examples

5.1. Host authentication

- The client sends an HTTP request with an Authorization header:

```
POST /some/endpoint HTTP/1.1
Host: photo.example
Date: Tue, 28 Aug 2012 09:41:21 -0400
Content-Type: application/x-www-url-form-encoded
Authorization: Dialback
               host="checkin.example",
               token="4430086d"

arg1=186&arg2=50
```

Figure 1

- The server checks that the Date: header is within an acceptable window (+/- 5 minutes recommended).

- The server checks that it hasn't seen this (host, url, token, date) tuple before.

- The server discovers a dialback confirmation endpoint at checkin.example. Its rel type is "dialback".

- The server posts an HTTP request to confirm the token:

```
POST /dialback HTTP/1.1
Host: checkin.example
Date: Tue, 28 Aug 2012 09:41:43 -0400
Content-Type: application/x-www-url-form-encoded

host=checkin.example&\
token=4430086d&\
url=http://photo.example/some/endpoint&\
date=Tue%2C%2028%20Aug%202012%2009%3A41%3A21%20-0400
```

Figure 2

Note: the "\" character is used here to indicate the line wrapping in the request content and is not part of the content itself.

- checkin.example returns 200 OK for a confirmation, 4xx for confirmation failure, 5xx for server failure.

5.2. Webfinger authentication

- The client sends an HTTP request with an Authorization header:

```
GET /some/resource HTTP/1.1
Host: photo.example
Date: Tue, 28 Aug 2012 09:41:21 -0400
Authorization: Dialback
               webfinger="alice@checkin.example",
               token="b3265cd5"
```

Figure 3

- The server checks that the Date: header is within an acceptable window (+/- 5 minutes recommended).

- The server checks that it hasn't seen this (webfinger, url, token, date) tuple before.

- The server discovers a dialback confirmation endpoint for alice@checkin.example. Its rel type is "dialback".

- The server posts an HTTP request to confirm the token:

```
POST /dialback HTTP/1.1
Host: checkin.example
Date: Tue, 28 Aug 2012 09:41:43 -0400
Content-Type: application/x-www-url-form-encoded

webfinger=alice@checkin.example&\
token=b3265cd5&\
url=http://photo.example/some/resource&\
date=Tue%2C%2028%20Aug%202012%2009%3A41%3A21%20-0400
```

Figure 4

Note: the "\" character is used here to indicate the line wrapping in the request content and is not part of the content itself.

- checkin.example returns 200 OK for a confirmation, 4xx for confirmation failure, 5xx for server failure.

6. IANA Considerations

6.1. Authentication scheme

This specification defines a new HTTP authentication scheme, "Dialback", per HTTP/1.1, part 7: Authentication [I-D.ietf-httpbis-p7-auth] to be registered at http://www.iana.org/assignments/http-authschemes.

- Authentication Scheme Name: Dialback
- Reference: (this document)

6.2. Link relation

This specification defines a new link relation type to be registered at http://www.iana.org/assignments/link-relations according to RFC 5988 [RFC5988].

- Relation Name: dialback
- Description: a dialback token verification endpoint
- Reference: (this document)

7. Security Considerations

7.1. Replay attacks

An attacker could capture the Authorization header of a request and replay the header for another payload.

To prevent replay attacks, the server MUST NOT accept a request if it has already seen a request with the same host or webfinger, url, token, and date.

Servers MAY mitigate storage requirements by rejecting requests with a Date: outside a fixed window. +/- 5 minutes from server time is reasonable.

7.2. Link discovery

An attacker could use DNS poisoning techniques to provide links to a false dialback endpoint.

Clients supporting dialback SHOULD support TLS for host-meta and Webfinger discovery.

HTTP servers SHOULD use the TLS endpoint for host-meta and Webfinger.

HTTP servers MAY fall back to the unencrypted equivalent.

7.3. Endpoint

An attacker could use DNS poisoning techniques to provide false responses to requests to the dialback verification endpoint.

HTTP clients support dialback SHOULD use TLS for dialback endpoints.

HTTP servers SHOULD require valid certificates for dialback endpoints.

7.4. Confidentiality

The dialback endpoint confirms that the host or Webfinger account is responsible for the HTTP request.

An attacker could use brute-force methods to determine if the host or Webfinger account has made an HTTP request to a given URL.

The dialback endpoint SHOULD NOT verify requests for dates outside a small window around the current time (+/- five minutes).

The dialback endpoint SHOULD use large enough tokens to make brute-force attacks impractical.

7.5. Data integrity

Dialback authentication does not confirm the contents of the HTTP request. For example, man-in-the-middle attack could replace the contents of a POST request with another payload, which would be verified.

Servers SHOULD use TLS to prevent man-in-the-middle attacks.

7.6. Denial of service

Dialback authentication lets the HTTP client induce the HTTP server to make additional verification requests.

By making requests with a host or webfinger parameter referring to a third party, a malicious client can cause extra HTTP requests to that third party.

To avoid denial-of-service attacks, HTTP servers SHOULD cache the results of host-meta and Webfinger requests.

8. References

8.1. Normative References

[I-D.ietf-httpbis-p7-auth]
Fielding, R., Lafon, Y., and J. Reschke, "HTTP/1.1, part 7: Authentication", draft-ietf-httpbis-p7-auth-20 (work in progress), July 2012.

[I-D.jones-appsawg-webfinger]
Jones, P., Salgueiro, G., and J. Smarr, "WebFinger", draft-jones-appsawg-webfinger-06 (work in progress), June 2012.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, October 2010.

[RFC6415] Hammer-Lahav, E. and B. Cook, "Web Host Metadata", RFC 6415, October 2011.

8.2. Informative References

[RFC5849] Hammer-Lahav, E., "The OAuth 1.0 Protocol", RFC 5849, April 2010.

Author's Address

Evan Prodromou
StatusNet, Inc.

Email: evan@status.net
URI: http://evan.status.net/