HTTPbis
Internet-Draft
Intended status: Standards Track
Expires: April 23, 2015

Yongming Zhao
Alibaba, Inc
Qinghuan Min
Alibaba, Inc
Xixi Xiang
Alibaba, Inc
Rui Chen
Alibaba, Inc

October 22, 2014

Hypertext Transfer Protocol: Access Control List
draft-zhao-http-acl-00

Abstract

In current Internet, HTTP/1.1 or HTTP/2 protocol has limited methods to control resource access. Usually it's achieved by modifying the configuration of the cache and proxy to enforce the access control rules. When original server's access control list (ACL) is updated, a reconfiguration of cache and proxy systems on the chain is required for the change to take effect, which always impacts the network stability. This document introduces a new access control mechanism, which introduces a new header field to dynamically control resource access.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on April 23, 2015.

Table of Contents

1. Introduction

In HTTP/1.1 or HTTP/2 protocol, there are few simple methods to control resource access. These methods might not work well with caches and proxies, which are widely used in current Internet.

Considering following case:

There some proprietary resources (html or image) in Web server, which should be accessed only by authorized requests. If the website is accelerated by CDN (Content Delivery Network), proprietary resources might be cached by servers in CDN. Thus access control should also be required on cache servers.

Practically it's neither possible nor efficient to have same authorization mechanism as original web server on all cache servers. HTTP 1.1 provides a simple solution by using "referrer" header field to validate incoming requests, which is widely used in CDN. But the solution has some drawbacks:

1. Inefficient with large scale networks

A proprietary resource could be cached publicly by cache or proxy after a legal request with proper “referrer” head was responded by original server, which means the designated access control is invalidated in this node. Following requests to cache or proxy with any “referrer” could access the resource. In order to protect the resource, access control should be enforced in each cache and proxy in the network. It’s troublesome and involving huge configuration work in a large scale networks with thousands of caches and proxies even with automation tool.

2. Inefficient with dynamic rule

The resource access control list may be updated frequently. The changed to access control list at original sever should be populated to caches and proxies in the network as soon as possible, or it means unexpected responses to user requests. Even with automation tool, there could be significant delay of populating in network with lots of caches and proxies. Practically, applying new access control might involve reloading configuration or even rebooting, which could have potential bad effect on network stability.

The goal in this proposal is to overcome the drawbacks mentioned and provide a better solution. A new header field is proposed to help cache and proxy to sync the access control rule from original server efficiently. The benefit of this solution increases as the number of cache and proxy in the network goes up. This method can be used in combination with HTTP/1.1 or HTTP/2.

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. Specification

The detail description of our approach as follow three sections:

Section 2.1 introduces a new header field X-REFERER-ACL. This field is only set in response header by original server. All original servers, caches and proxies MUST check the request with certain rule.

Section 2.2 introduces the mapping rules of X-REFERER-ACL. We will give some examples to show the checking routine.

Section 2.3 describes rule of how to handle the cache expires time.

2.1. HTTP header field extension

A new header field named “X-Referer-ACL” is introduced. This header field MUST be only set in response header, if it is set in request header, original server, cache or proxies MUST ignore it.

Here’s the format of the new header field:

X-Referer-ACL = X-Referer-ACL: ACTION TYPE PARAMS

ACTION = "A" | "D"

A = Allow ALL

D = Deny ALL

TYPE = "*" | "1" | "2"

* = Matching ALL

1 = Matching Domain

2 = Matching Hostname

PARAMS=","token

Description:

1. ACTION is a flag which indicates allow or deny. This value MUST be one of the upper case “A” or “D”.

   “A” means Allow ALL. We can also consider it as “white list”.

   “D” means Deny ALL. We can also consider it as “black list”.

2. TYPE value is a single character, it indicates the mapping type.

   * means to check all type.

   1 means to check domain name. It MUST check only domain name of referer field of request.

   2 means to check hostname name. It MUST check only hostname of referer field of request.

3. PARAMS is parameter list. The item of PARAMS is hostname or domain name, which is determined by “TYPE” value. The list item must be separated by comma.

   How to set the value:

   If TYPE=”*”, it means no matter hostname or domain name will be checked. PARAMS is optional and should be ignored.

If TYPE="1", domain name checking. PARAMS is the list of domain name which is allowed or denied to access the resource. Domain name MUST not has (.) in the headmost.

If TYPE="2", hostname checking. RARAMS is the list of hostname list which is allowed or denied to access the resource.

4. X-Referer-ACL may include several ACTION TYPE PARAMS items separated by semi-colon (;). ACTION TYPE PARAMS items should be match one by one in order.

Here is an example:

```
X-Referer-ACL: A 1 A.B.taobao.com; D 1 B.taobao.com; A 1 taobao.com, taobaocdn.com; D*
```

The meaning of the header field value above is: allow A.B.taobao.com, taobao.com and taobaocdn.com and their child domain name access the resource. Deny B.taobao.com and any other domain name access it.

2.2. Matching rule of X-Referer-ACL field

No matter caches or proxies, when receiving the X-Referer-ACL header field in response, they MUST observe the rule as following:

1. If there is no X-Referer-ACL field in the response, caches and proxies should allow all request to access the resource and no need to do any check.

2. If there is no referer in request, caches and proxies should not check the request, even if response has the X-Referer-ACL field.

3. If request has referer field and response has X-Referer-ACL, caches and proxies MUST check rule of X-Referer-ACL field and decide whether to allow or deny current request.

4.If refer doesn't match any X-Refer-
ACL rule, the request SHOULD be allowed unconditionally.

There are some more notices for our matching rule:

a) X-Referer-ACL field MUST appear only once. It is for simplifying the macthing rule.

b) If there are more than one parameter list matching the request, the first mapping value's ACTION should be used.

c) If the first ACTION TYPE PARAMS is D*, all requests should be rejected. If the first ACTION TYPE PARAMS is A*, all requests should be allowed.

d) If referer is not set in the request or the value of this field is empty, original server, cache and proxies will allow this request unconditionally.

e) If referer's value is not a correct URI (not http or https), the request should be denied with HTTP 403.

f) Hostname and domain name would be normalized to lower case before matching procedure.

We believe that the new field introduced in section 2.1 and the rule defined in section 2.2 would help to enforce access control in large scale network more efficiently.

2.3. How to handle the cache expiration time

If the resource being requested in the cache server is already expired, the cache server MUST send a GET request with "If-Modify-Since" to original server to check whether the resource is changed or not. Original server should return HTTP 200 if the resource is changed. HTTP 304 Not Modified should be returned if the resource is untouched and the X-Referer-ACL MUST be included in the response header.

3. Security Considerations

The new field introduced in this proposal is related with anti-theft and access control. Cache server MUST record the X-Referer-ACL rule of response. X-Referer-ACL MUST set in header of response every time as HTTP request has no status. Once response doesn't have the X-Referer-ACL at any time, the cache system will remove the old ACL rule immediately, which will invalidate the access control and expose the resource to the risk of unauthorized access.

Cache server MUST not change any of the X-Referer-ACL values because any change would make the original server access control invalidated. Cache servers and proxies MUST follow section 2.1 and section 2.2 to check the entire request. Failing to follow might access control invalidation.

4. IANA Considerations

4.1. Header Field Registration

HTTP header fields are registered within the Message Header Field Registry maintained at [1].

This document defines the following HTTP header fields, so their associated registry entries shall be updated according to the permanent registrations below (see [BCP90]):

Header Field Name	Protocol	Status	Reference

| X-Referer-ACL | http | proposed standard | Section 2.1 |

+-------------------+----------+-------------------+-------------+

The change controller is: "IETF (iesg@ietf.org) - Internet Engineering Task Force".

5. References

5.1. Normative References

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

[2] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

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

[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

5.2. Informative References

[3] Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.

[Fab1999] Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.

Authors' Addresses

Yongming Zhao

0825, BUILDING North 4, ZHONGHONG BEIJINGXIANGSU, #1 WULIQIAOYIJIE

CHANGYING, CHAOYANG, BEIJING

China

Email: zym@efengcloud.com

QingHuan Min
Alibaba
No.1 East 3rd Ring Middle Rd
Chaoyang District, Beijing
China

Email: zongyi.mqh@taobao.com

XiXi Xiang
Alibaba
HangZhou, ZheJiang province,
China

Email: xixi.xxx@alibaba-inc.com

Rui Chen (editor)
Alibaba
No.1 East 3rd Ring Middle Rd
Chaoyang District, Beijing
China

Email: zhuquan.cr@taobao.com