Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 19, 2011

M. Nottingham
August 18, 2010

The Network Authentication Required HTTP Status Code
draft-nottingham-http-portal-01

Abstract

"Captive portals" are a commonly-deployed means of obtaining access credentials and/or payment for a network. This memo introduces a new HTTP status code as a means of addressing issues found in these deployments.

This memo should be discussed on the ietf-http-wg@w3.org mailing list, although it is not a work item of the HTTPbis WG.

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 February 19, 2011.

Copyright Notice

Table of Contents

1. Introduction

It has become common for networks to require authentication, payment and/or acceptance of terms of service before granting access. Typically, this occurs when accessing "public" networks such as those in hotels, trains, conference centres and similar networks.

While there are several potential means of providing credentials to a network, these are not yet universally supported, and in some instances the network administrator requires that information (e.g., terms of service, login information) be displayed to end users.

In such cases, it has become widespread practice to use a "captive portal" that diverts HTTP requests to the administrator's web page. Once the user has satisfied requirements (e.g., for payment, acceptance of terms), the diversion is ended and "normal" access to the network is allowed.

Typically, this diversion is accomplished by one of several possible techniques;

- IP interception - all requests on port 80 are intercepted and send to the portal.
- HTTP redirects - all requests on port 80 are intercepted and an HTTP redirect to the portal's URL is returned.
- DNS interception - all DNS lookups return the portal's IP address.

In each case, the intent is that users connecting to the network will open a Web browser and see the portal.

However, because port 80 is used for non-browser traffic, a number of issues (see Appendix B) have been encountered.

This memo introduces a new HTTP status code, 428 Network Authentication Required, as a solution to some of these issues. Appendix A outlines how it might be used in typical deployments.

2. 428 Network Authentication Required

This status code indicates that the client should authenticate to gain network access before resubmitting the request.

The associated representation SHOULD indicate how to do this; e.g., with an HTML form for submitting credentials.

Responses with the 428 status code MUST NOT be stored by a cache.

3. Security Considerations

In common use, a response carrying the 428 status code will not come from the origin server indicated in the request's URL. This presents many security issues; e.g., an attacking intermediary may be inserting cookies into the original domain's name space, may be observing cookies or HTTP authentication credentials sent from the user agent, and so on.

However, these risks are not unique to the 428 status code; in other words, a captive portal that is not using this status code introduces the same issues.

4. IANA Considerations

The HTTP Status Codes Registry should be updated with the following entry:

- Code: 428
- Description: Network Authentication Required
- Specification: [this document]

Appendix A. Using the 428 Status Code

This appendix demonstrates a typical use of the 428 status code; it is not normative.

A network operator wishing to require some authentication, acceptance of terms or other user interaction before granting access usually does so by identify clients who have not done so ("unknown clients") using their MAC addresses.

Unknown clients then have all traffic blocked, except for that on TCP port 80, which is sent to a HTTP server (the "login server") dedicated to "logging in" unknown clients, and of course traffic to the login server itself.

For example, a user agent might connect to a network and make the following HTTP request on TCP port 80:

```
GET /index.htm HTTP/1.1
Host: www.example.com
User-Agent: ExampleAgent
```

Upon receiving such a request, the login server would generate a 428 response:


```
HTTP/1.1 428 Network Authentication Required
Refresh: 0; url=https://login.example.net/
Content-Type: text/html

<html>
   <head>
   </head>
   <body>
      <h1>You are being redirected to log into the network...</h1>
   </body>
</html>
```

Here, the 428 status code assures that non-browser clients will not interpret the response as being from the origin server, and the Refresh header redirects the user agent to the login server (an HTML META element can be used for this as well).

Note that the 428 response can itself contain the login interface, but it may not be desirable to do so, because browsers would show the login interface as being associated with the originally requested URL, which may cause confusion.

Appendix B. Issues Raised by Captive Portals

Since clients cannot differentiate between a portal's response and that of the HTTP server that they intended to communicate with, a number of issues arise.

One example is the "favicon.ico" <http://en.wikipedia.org/wiki/Favicon> commonly used by browsers to identify the site being accessed. If the favicon for a given site is fetched from a captive portal instead of the intended site (e.g., because the user is unauthenticated), it will often "stick" in the browser's cache (most implementations cache favicons aggressively) beyond the portal session, so that it seems as if the portal's favicon has "taken over" the legitimate site.

Another browser-based issue comes about when P3P <http://www.w3.org/TR/P3P/> is supported. Depending on how it is implemented, it's possible a browser might interpret a portal's response for the p3p.xml file as the server's, resulting in the privacy policy (or lack thereof) advertised by the portal being interpreted as applying to the intended site. Other Web-based protocols such as WebFinger <http://code.google.com/p/webfinger/wiki/WebFingerProtocol>, CORS <http://www.w3.org/TR/cors/> and OAuth <http://tools.ietf.org/html/draft-ietf-oauth-v2> may also be

vulnerable to such issues.

Although HTTP is most widely used with Web browsers, a growing number of non-browsing applications use it as a substrate protocol. For example, WebDAV <http://tools.ietf.org/html/rfc4918> and CalDAV <http://www.ietf.org/rfc/rfc4791.txt> both use HTTP as the basis (for network filesystem access and calendaring, respectively). Using these applications from behind a captive portal can result in spurious errors being presented to the user, and might result in content corruption, in extreme cases.

Similarly, other non-browser applications using HTTP can be affected as well; e.g., widgets <http://www.w3.org/TR/widgets/>, software updates, and other specialised software such as Twitter clients and the iTunes Music Store.

It should be noted that it's sometimes believed that using HTTP redirection to direct traffic to the portal addresses these issues. However, since many of these uses "follow" redirects, this is not a good solution.

Appendix C. Non-HTTP Applications and Techniques

This memo does not address non-HTTP applications, such as IMAP, POP, or even TLS-encapsulated HTTP. Since captive portals almost always target Web browsers (has anyone ever seen one that inserts an e-mail into your inbox asking you to authenticate?), this is appropriate.

Instead, it is anticipated that well-behaved portals will block all non-HTTP ports (especially port 443) until the user has successfully authenticated.

Overall, there may also be an interesting discussion to be had about improving network access methods to the point where a user interface can be presented for the same purposes, without resorting to intercepting HTTP traffic. However, since such a mechanism would by necessity require modifying the network stack and operating system of the client, this memo takes a more modest approach.

Appendix D. Acknowledgements

The author takes all responsibility for errors and omissions.

Author's Address

Mark Nottingham

Email: mnot@mnot.net
URI: http://www.mnot.net/