

Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 6, 2016

M. Nottingham
April 4, 2016

Captive Portals Problem Statement
draft-nottingham-capport-problem-01

Abstract

This draft attempts to establish a problem statement for "Captive Portals", in order to inform discussions of improving their operation.

Note to Readers

The issues list for this draft can be found at
<https://github.com/mnot/I-D/labels/capport-problem> .

The most recent (often, unpublished) draft is at
<https://mnot.github.io/I-D/capport-problem/> .

Recent changes are listed at <https://github.com/mnot/I-D/commits/gh-pages/capport-problem> .

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 October 6, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Notational Conventions	2
2. Defining Captive Portals and Networks	2
2.1. Why Captive Portals Are Used	3
3. Issues Caused by Captive Portals	3
4. Issues Caused by Captive Portal Detection	5
4.1. Issues Caused by Defeating Captive Portal Detection . . .	5
5. Security Considerations	5
6. References	6
6.1. Normative References	6
6.2. URIs	6
Appendix A. Acknowledgements	6
Author's Address	6

1. Introduction

This draft attempts to establish a problem statement for "Captive Portals", in order to inform discussions of improving their operation.

1.1. Notational Conventions

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 [RFC2119].

2. Defining Captive Portals and Networks

A "_captive network_" is a network that employs a captive portal for a variety of purposes; see Section 2.1. A "_captive portal_" is a Web site that captive networks direct users to.

This is achieved by directing requests for "normal" Web access to the captive portal, through variety of techniques, including DNS poisoning, TCP interception, HTTP response modification and/or HTTP redirection.

Once the captive network's goals are met, the network "remembers" that the user is allowed network access, usually by MAC address, although there is a significant amount of variance between implementations.

Over time, operating systems have developed "_captive portal detection_" processes to discover captive networks, and to assist users through the process of obtaining full network access. They often involve specialised "_captive portal browsers_" which only allow the captive portal to use a subset of the full capabilities of a Web browser, and have a different user experience.

2.1. Why Captive Portals Are Used

Captive portals are deployed in a variety of situations, but the most common motivations are:

- o *Authentication* - Obtaining user credentials before authorising network access
- o *Payment* - Obtaining payment for using the network.
- o *Information* - Presenting information to the user. This might include displaying legal notices, details about the network provider and/or its location, advertisements, policies, etc., and obtaining user consent.
- o *Notifications* - Some networks use the same mechanisms as captive portals to notify users of account status, network downtime, emergency alerts, etc. See [RFC6108] for an example of one way this is done.

In all of these cases, using a Web browser is attractive, because it gives the network the ability to tailor the user's interface and experience, as well as the ability to integrate third-party payment, advertising, authentication and other services.

3. Issues Caused by Captive Portals

When a network imposes a captive portal, it can cause a variety of issues, both for applications and end users.

- o *False Negatives* - Because so many different heuristics are used to detect a captive portal, it's common for an OS or browser to think it's on an open network, when in fact there is a captive portal [4] in place.

- o **Longevity** - Often, it's necessary to repeatedly log into a captive portal [5], thanks to timeout issues. The effects of this range from annoyance to inability to complete tasks, depending on the timeout and the task at hand.
- o **Interoperability Issues** - Captive portals often depend on specific operating system and browser capabilities and behaviours. Client systems that do not share those quirks often have difficulty connecting to captive portals.
- o **Confusion** - Because captive portals are effectively a man-in-the-middle attack, they can confuse users as well as user agents (e.g., caches). For example, when the portal's TLS certificate doesn't match that of the requested site, or the captive portal's /favicon.ico gets used as that of the originally requested site.
- o **DNS*/*DNSSEC** - When portals respond with forged DNS answers, they confuse DNS resolvers and interoperate poorly with host-validating DNSSEC resolvers and applications.
- o **TLS** - Portals that attempt to intercept TLS sessions (HTTPS, IMAPS, or other) can cause certificate error messages on clients, encouraging bad practice to click through such errors.
- o **Unexpected Configuration** - Some captive portals do not work with clients using unexpected configurations, for example clients using static IP, custom DNS servers, or HTTP proxies.
- o **Stealing Access** - because captive portals often associate a user with a MAC address, it is possible for an attacker to impersonate an authenticated client (e.g., one that has paid for Internet access). Note that this is specific to open Wifi, and can be prevented by using a secure wireless medium. However, configuration of secure wireless is often deemed to be too complex for captive networks.
- o **Non-Browser Clients** - It is becoming more common for Internet devices without the ability to run a browser to be used, thanks to the "Internet of Things." These devices cannot easily use most networks that interpose a captive portal.
- o **Connectivity Interruption** - For a device with multiple network interfaces (e.g., cellular and WiFi), connecting to a network can require dropping access to alternative network interfaces. If such a device connects to a network with a captive portal, it can lose network connectivity until the captive portal requirements are satisfied.

4. Issues Caused by Captive Portal Detection

Many operating systems attempt to detect when they are on a captive network. Detection aims to minimize the negative effects caused by interposition of captive portals, but can cause different issues, including:

- o **False Positives** - Some networks don't use a Web browser interface to log in; e.g., they require a VPN to access the network [6], so captive portal detection relying on HTTP is counterproductive.
- o **Non-Internet Networks** - Some applications [7] and/or networks don't assume Internet access, but captive portal detection often conflates "network access" with "Internet access".
- o **Sandboxing** - When a captive portal is detected, some operating systems access the captive portal in a highly sandboxed captive portal browser. This might have reduced capabilities, such as limited access to browser APIs. In addition, this environment is separate from a user's normal browsing environment and therefore does not include state. While sandboxing seems a good idea to protect user data (particularly on Open WiFi), it is implemented differently on various platforms and often causes a (severely) broken user experience on the captive portal (even when the operator is protecting user data end-to-end with HTTPS). To offer a consistent and rich experience on the captive portal, some operators actively try to defeat operating system captive portal detection.

4.1. Issues Caused by Defeating Captive Portal Detection

Many captive portal devices provide optional mechanisms that aim to defeat captive portal detection.

Such defeat mechanisms aim to avoid the problems caused by captive portal detection (see Section 4), with the consequence that they also cause the same problems that detection was intended to avoid (see Section 3).

5. Security Considerations

TBD

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6108] Chung, C., Kasyanov, A., Livingood, J., Mody, N., and B. Van Lieu, "Comcast's Web Notification System Design", RFC 6108, DOI 10.17487/RFC6108, February 2011, <<http://www.rfc-editor.org/info/rfc6108>>.

6.2. URIs

- [1] <https://discussions.apple.com/thread/6251349>
- [2] https://community.aerohive.com/aerohive/topics/ios_7_captive_portal_issues
- [3] <http://stackoverflow.com/questions/14606131/using-captive-network-assistant-on-macosx-to-connect-to-vpn>
- [4] <http://forum.piratebox.cc/read.php?9,8879>
- [5] <https://github.com/httpwg/wiki/wiki/Captive-Portals>

Appendix A. Acknowledgements

This draft was seeded from the HTTP Working Group Wiki Page on Captive Portals [8]; thanks to all who contributed there.

Thanks to Martin Thomson, Yaron Sheffer, David Bird and Jason Livingood for their suggestions.

Author's Address

Mark Nottingham

Email: mnot@mnot.net

URI: <https://www.mnot.net/>