Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 8, 2021

R. Pietrak

February 4, 2021

Cookie Extention Limitting Its Availability to User Agent Components
draft-pietrak-cookie-scope-00

Abstract

This memo addresses cookies security by introduction of an additional constraints, web application designer may impose on cookie availability for localhost applications components.

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 https://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 August 8, 2021.

Table of Contents

1. Introduction

As of current standard, HTTP state management mechanism RFC 6265 [refs.RFC6265], provide tools (in the form of cookie attributes) to limit the dissemination of any particular cookie. Those constraints are:

- Cookie domain
- Cookie expiration time
- HttpOnly attribute
- SecureOnly attribute

But, current standard doesn't provide means to control the scope cookies are available to localhost components, like user agent software. These days search engines provide host of evidence that programmers continuesly look for such means. Usually, the question is: "How do I limit my session cookie to just a single tab?".

The questions comes from evident need to move away out off user session-ID (acquired after login/pass veryfication) visible in plain sight as part of URL referring to the access limitted pages. Usually such modyfications are evaluated under strict requirement, that current functionality remains unaltered. Here the required functionality usually boils down to having such session-ID not shared

among interface components which waren't involved in credencials veryfication. Meaning, that only the tab that provided credencials (login/pass) will be authorised to access those particular resources. In other words: since currently every window and every tab may have a diffrent URL retrieved, it should also be possible for it to have different (from other tabs) session login credencials.

This is considered a desired (even required) feature.

As of now, storing such login session credencials within a Cookie (which is also whidely used to hold session credencials) alters this functionality, and so is not acceptable for those migrations. The functionality is lost, because a cookie set in one tab of web browser will be immediately available to all other tabs.

On the other hand, current functionality of sharing all cookies between all application components is desired by other web application programmers. This memo introduces cookie attribute, which will maintain current cookie behavior, while allowing for currently missing cookie scope limitations to be defined by web application programmer.

2. Terminology

The following terms will be used in this memo:

user: Within a computer system the term "user" means any set of components holding the same security credencials. These credencials may originate (or be derived) from a physical person giving them out to a login application, or they may originate from kernel mainteined vault of credencials designated for system components when system (unattended) application actions are due. System services like ntp service, or MTA (mail transport agent) are examples of the later case.

application: Any programatic component of a computer system.

window: Any input/output channel, that an application may have an access to independently from other such channels. An example of such channel is GUI window, which gets data (like graphics) from an application, while all other windows of that same application does not get disturbed by that data stream. Every HTTP datastream, from open to close is considered a separate window, with the exeption below, where "tabs" are defined.

window-tab: Any part of a window, that holds and presents (or processes) one HTML document, including all objects that may be retrieved separately (separate HTTP connections), but are a part

(functional or aestetical) of the main document. It's valid to say, that a tab is whatever is necesary to present (or processe) whatever server returns in response for a single user click, including all the subcomponents, that HTML defines to be retrieved as a result of that click.

tab: synonim to window-tab

viewport: A single window-tab currently selected (activated/focused/visible) to/by the user.

3. CookieRadius attribute

CookieRadius attribute can have only one of the following four values, and system behavior for each of them is the following:

World: Cookie will be available to all user applications. Every web browsers launched by a particular user will see that cookie.

Windows: Cookie will be available to all the windows of an application that received that cookie.

Tabs: Cookie will be available to all the tabs of the same window. A window, a tab that received that cookie, belongs to.

Viewport: Cookie will not be available to any other system component, but the tab, that received it.

When CookieRadius is not defined by a cookie, it MUST be assumed to have a value of "Windows". When CookieRadius is defined, but it's value is unrecognized, applications MUST assume it's value is "Viewport".

HttpOnly cookie attribute is completely independent and implementations WILL NOT correlate values of HttpOnly attribute with any value of CookieRadius attribute.

Cookie "domain" interferes with CookieRadius only when its value is "Viewport". In that case (either explicitly set or assumed as default), "domain" is set to domain of URL retrieved irrespectively of setting within the cookie. Consequently availability of such cookie is not only limitted to a single viewport, but also to a "domain" the tab content originated from. In other words, "Viewport" cookies never traverse domains.

One notable consequence of the above "domain" restriction is the fact, that no other component (like an embeded picture from a different "domain") will ever get any information of user getting

logged-in to any particular "main-tab-domain". This is a security feature.

User agent MAY let users further tighten the scope of a cookie below the radius declared in CookieRadius attribute, but it MUST NOT allow user to expand the radius. That is particularly important for "HttpOnly=false" cookies.

4. Security

The actual impact of the proposed cookie attribute can only be truelly evaluated after its wide implementation and use in other then here presented scenarios. The potencial to limit the leackage of security data (login credencials) between application components may help improve internet security.

5. References

[refs.RFC6265]
Barth, A., "HTTP State Management Mechanism", RFC 6265, April 2011.

Author's Address

Rafal Pietrak

Email: cookie.rp@ztk-rp.eu