

Is there a browser equivalent to IE's ClearAuthenticationCache?

Asked 16 years, 3 months ago Modified 10 months ago Viewed 47k times



I have a few internal .net web application here that require users to "log out" of them. I know this may seem moot on an Intranet application, but nonetheless it is there.

27



We are using Windows authentication for our Intranet apps, so we tie in to our Active Directory with Basic Authentication and the credentials get stored in the browser cache, as opposed to a cookie when using .net forms authentication.



In IE6+ you can leverage a special JavaScript function they created by doing the following:

```
document.execCommand("ClearAuthenticationCache", "false")
```

However, for the other browsers that are to be supported (namely Firefox at the moment, but I strive for multi-browser support), I simply display message to the user that they need to close their browser to log out of the application, which effectively flushes the application cache.

Does anybody know of some commands/hacks/etc. that I can use in other browsers to flush the authentication cache?

asp.net

authentication

cross-browser

Share

Improve this question

Follow

edited Sep 26, 2008 at 19:56



Chris

6,842 ● 6 ● 53 ● 67

asked Aug 27, 2008 at 23:34



Dillie-O

29.7k ● 14 ● 102 ● 141

6 Answers

Sorted by: Highest score (default)



I've come up with a fix that seems fairly consistent but is hacky and [I'm still not happy with it.](#)

13

It does work though :-)



1) Redirect them to a Logoff page



2) On that page fire a script to ajax load another page with dummy credentials (sample in jQuery):

```
$j.ajax({
  url: '<%:Url.Action("LogOff401", new { id = random })%>',
  type: 'POST',
  username: '<%:random%>',
  password: '<%:random%>',
  success: function () { alert('logged off'); }
});
```

3) That should always return 401 the first time (to force the new credentials to be passed) and then only accept the dummy credentials (sample in MVC):

```
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult LogOff401(string id)
{
    // if we've been passed HTTP authorisation
    string httpAuth = this.Request.Headers["Authorization"];
    if (!string.IsNullOrEmpty(httpAuth) &&
        httpAuth.StartsWith("basic", StringComparison.OrdinalIgnoreCase))
    {
        // build the string we expect - don't allow regular users to pass
        byte[] enc = Encoding.UTF8.GetBytes(id + ':' + id);
        string expected = "basic " + Convert.ToBase64String(enc);

        if (string.Equals(httpAuth, expected,
            StringComparison.OrdinalIgnoreCase))
        {
            return Content("You are logged out.");
        }
    }

    // return a request for an HTTP basic auth token, this will cause XmlHttp
    to pass the new header
    this.Response.StatusCode = 401;
    this.Response.StatusDescription = "Unauthorized";
    this.Response.AppendHeader("WWW-Authenticate", "basic realm=\"My Realm\"");

    return Content("Force AJAX component to sent header");
}
```

4) Now the random string credentials have been accepted and cached by the browser instead. When they visit another page it will try to use them, fail, and then prompt for the right ones.

Share

Improve this answer

Follow

edited May 23, 2017 at 12:25



Community Bot

1 • 1

answered Jun 9, 2011 at 7:49



Keith

155k • 82 • 306 • 446

Hmm, that is a rather interesting approach to things. Can you do it without MVC if I already have WinForms apps in place? – [Dillie-O](#) Jun 9, 2011 at 18:58

@Dillie-O yeah, just create a new `.aspx` or `.ashx` page, stick this in the page load, add `string id = Request["id"];` to the top and replace the two `return Content(...` with `Response.End()`. Alternatively you could create a custom `HttpHandler`. All it need to do is return an HTTP 401 for the valid username & password, and then return an HTTP 200 for the dummy one (which fools the browser into caching that instead). – Keith Jun 9, 2011 at 22:14

- 2 This works reliably but on FF and WebKit it will bring up a login dialog that might confuse the user. I was hoping that the use of AJAX would prevent that. Still researching. – AnthonyVO Dec 14, 2011 at 20:54

@AnthonyVO - I wrote this originally on Chrome (webkit based) and it didn't re-prompt, I'm not sure why that's different for you. I've only included a subset of my code though, so it's possible I missed something in my answer. What tech stack are you using? – Keith Dec 15, 2011 at 0:28

IIS 7.5 plus all browsers. Not MVC WebApp. Regarding your comment to @Dillie-O about using a `HttpHandler`, that won't work because a `HttpHandler` won't see the authentication Traffic. See: stackoverflow.com/questions/769432/ihttphandler-vs-ihttpmodule. Going to try a `iHttpModule`.. – AnthonyVO Dec 15, 2011 at 21:39



A couple of notes. A few people have said that you need to fire off a ajax request with invalid credentials to get the browser to drop it's own credentials.

7



This is true but as Keith pointed out, it is essential that the server page claims to accept these credentials for this method to work consistently.



On a similar note: It is NOT good enough for your page to just bring up the login dialog via a 401 error. If the user cancels out of the dialog then their cached credentials are also unaffected.

Also if you can please poke MOZILLA at https://bugzilla.mozilla.org/show_bug.cgi?id=287957 to add a proper fix for FireFox. A webkit bug was logged at https://bugs.webkit.org/show_bug.cgi?id=44823. IE implements a poor but functional solution with the method:

```
document.execCommand("ClearAuthenticationCache", "false");
```

It is unfortunate that we need to go to these lengths just to log out a user.

Share

edited Dec 14, 2011 at 14:48

answered Dec 13, 2011 at 23:39

Improve this answer



AnthonyVO

3,973 ● 1 ● 42 ● 41

Follow



Mozilla implemented the `crypto` object, available via the DOM `window` object, which has the `logout` function (Firefox 1.5 upward) to clear the SSL session state at the

7

browser level so that "the next private operation on any token will require the user password again" (see [this](#)).



The crypto object seems to be an implementation of the [Web Crypto API](#), and according to [this document](#), the DOMCrypt API will add even more functions.



As stated above Microsoft IE (6 upward) has:

```
document.execCommand("ClearAuthenticationCache", "false")
```

I have found no way of clearing the SLL cache in Chrome (see [this](#) and [this](#) bug reports).

In case the browser does not offer any API to do this, I think the better we can do is to instruct the user to close the browser.

Here's what I do:

```
var agt=navigator.userAgent.toLowerCase();
if (agt.indexOf("msie") !== -1) {
    document.execCommand("ClearAuthenticationCache","false");
}
//window.crypto is defined in Chrome, but it has no logout function
else if (window.crypto && typeof window.crypto.logout === "function"){
    window.crypto.logout();
}
else{
    window.location = "/page/to/instruct/the/user/to/close/the/browser";
}
```

Share

edited Feb 25, 2012 at 15:55

answered Feb 25, 2012 at 12:29

Improve this answer



staromeste

931 ● 7 ● 5

Follow

- 2 instead of checking the userAgent string, you should check if the command is supported by using:

```
if document.queryCommandSupported("ClearAuthenticationCache") { ... }
```

 – [Bart Jolling](#) May 4, 2017 at 13:09



I've been searching for a similar solution and came across a patch for Trac (an issue management system) that does this.

3



I've looked through the code (and I'm tired, so I'm not explaining everything); basically you need to do an AJAX call with **guaranteed invalid** credentials **to your login page**. The browser will get a 401 and know it needs to ask you for the right credentials next time you go there. You use AJAX instead of a redirect so that you can specify incorrect credentials and the browser doesn't popup a dialog.



On the patch (<http://trac-hacks.org/wiki/TrueHttpLogoutPatch>) page they use very rudimentary AJAX; something better like jQuery or Prototype, etc. is probably better, although this gets the job done.

Share Improve this answer Follow

answered Sep 10, 2009 at 19:03



Marius Marais

1,086 ● 7 ● 7

This works 100% and does not impact your browser history – [Kevin Potgieter](#) Mar 10, 2020 at 13:13



2

Why not use FormsAuth, but against ActiveDirectory instead as per the info in [this thread](#). It's just as (in)secure as Basic Auth, but logging out is simply a matter of blanking a cookie (or rather, calling [FormsAuthentication.SignOut](#))



Share

Improve this answer



Follow



edited May 23, 2017 at 11:46



Community Bot

1 ● 1

answered Aug 28, 2008 at 0:10



Duncan Smart

32k ● 11 ● 70 ● 72

I could look into the FormsAuth thing, but right now the policy is to use Windows authentication against the system. I'll check into all three of these options. If anybody else thinks of something, by all means post it. – [Dillie-O](#) Aug 28, 2008 at 14:58



0

Well, I've been browsing around Bugzilla for a bit now and seemingly the best way you can go for clearing the authentication would be to send non-existent credentials.

Read more here: https://bugzilla.mozilla.org/show_bug.cgi?id=287957



Share Improve this answer Follow

answered Aug 27, 2008 at 23:44



Andy

250 ● 1 ● 2

