

## Re: Major security issue?

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2004-02/0031.html>

---

**From:** Paul Glavich (*glav\_at\_aspalliance.com-NOSPAM*)

**Date:** 02/01/04

Date: Sun, 1 Feb 2004 22:06:19 +1100

I replied to this on the other list, but thought I'd send it here as well.

We have used cookieless sessions and what you say is true, but we used SSL to encrypt traffic, which as you know requires a connection to the same client/server (ie. if connection broken, then the SSL session is invalid) so this IP verification approach could still work but it assumes SSL, which of course is really outside of ASP.NET's domain.

Further to this you could use client certs to verify integrity which strictly doesn't stop people from hijacking a session (simply minimises it), but there are just som many ways to approach this, each with positives and negatives, that if the ASP.NET team adopted one approach, it would be implicitly be advocating this one approach which may very well be flawed under a number of different situations.

My 2 cents.

--

- Paul Glavich

"Keith" <keith@keithadler.com> wrote in message  
news:7c0f01c3e881\$00918d70\$a001280a@phx.gbl...

> This is predictable in an insecure product. I'm not  
> trying to act as if I discovered something new or Earth  
> shattering, but I am quite surprised there is nothing in  
> place in ASP.Net to protect user sessions from being  
> hijacked. It seems to me that the session IDs have been  
> problematic since ASP first came about. In ASP.Net they  
> are still for some reason handed out in a fashion that  
> means the same ID could be sent out to the same browser  
> even after a Session.Abandon(). It doesn't make sense  
> that Microsoft couldn't do something as simple as encrypt  
> the user agent and source IP into the session GUID if the  
> user wanted to lock the source and device of a request  
> down to a particular computer/network. From an  
> architectural standpoint I realize that this in itself  
> would add some overhead to IIS because every HTTP request  
> would have to be checked against a lookup, but with HTTP  
> keep-alives this check would only need to occur once on  
> the same connection. I also realize that someone could  
> use this to DoS a server by sending lots of HTTP requests  
> with random IDs that would have to decoded and matched up  
> against connections, but I'm sure that intrusion

## microsoft.public.dotnet.framework.aspnet: Re: Major security issue?

> detection systems could be made to deal with this issue.  
> The other option of course is to not use cookieless  
> sessions under the anonymous user configuration and rely  
> on an in-memory cookie which is obviously a little less  
> accessible. In either situation though, this seems like  
> an incredible option to not provide ASP users.  
>  
> >-----Original Message-----  
> >It seems to me that this would be listed as a  
> predictable downside to using  
> >cookieless sessions. Verifying IPs and/or user agents  
> wouldn't be any real  
> >way to avoid this, so it makes sense to me that this  
> wouldn't be the default  
> >behavior for asp.net to check that. And if it were to  
> check it, where would  
> >it store this info? In session variables? Hmmph.  
> >  
> >--  
> >  
> >Ray at home  
> >Microsoft ASP MVP  
> >  
> >  
> >"Keith" <keith@keithadler.com> wrote in message  
> >news:77b301c3e87d\$0ff55c00\$a101280a@phx.gbl...  
> >> I have found what I believe to be a serious security  
> >> issue in ASP.Net. If you have:  
> >>  
> >> 1. Your website configured for anonymous access  
> >> 2. Elect under web.config to set the sessionstate  
> >> attribute of cookieless to true  
> >>  
> >> Anyone from any IP address or across another browser  
> > can  
> >> copy the URL and work within the session. My question  
> >> is "Why doesn't ASP.Net provide an option around  
> > ensuring  
> >> all requests for a user session originate from the same  
> >> IP address and/or same useragent?" I know that some  
> >> people sit behind firewalls, proxies and layer 4  
> > devices  
> >> that could load balance and affect HTTP traffic, but it  
> >> honestly escapes me why I can access my web application  
> >> on any machine inside or outside of my network with  
> > just  
> >> the sessionid in the URL from even different browsers.  
> >> There must be a way to control this in the  
> >> configuration. Am I alone in find this troubling?  
> >  
> >  
> >.  
> >