

## Re: Major security issue?

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2004-02/0026.html>

---

**From:** Ray at [MVP] (%=sLocation%)

**Date:** 02/01/04

Date: Sun, 1 Feb 2004 04:33:04 -0500

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

> *This is predictable in an insecure product.*

I can loosen all your lugnuts with a standard crossbar wrench, too. Does that make your car an insecure product? If you think it does, use >=3 wheel locks on each of your wheels.

> *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.*

There is, real sessions, although that is arguable as well. And this has nothing to do with what server-side technology you choose to use.

> *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().*

Even if this happened, would it matter? It'd still be a new session.

> *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.*

That data is meaningless though. When you have 1000 computers created from the same image all sitting behind the same firewall, for example.

> *From an*

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

- > *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 be decoded and matched up*
- > *against connections, but I'm sure that intrusion*
- > *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.*

Cookieless sessions are just an alternative. If you're that worried about them, don't use them. This is not a design flaw in ASP; this is just a result of the technology that you're using and the way it works. If you choose to use querystrings to identify users, it doesn't matter what kind of server-side technology you use if you're catering to the cookie-paranoid people.

- > *In either situation though, this seems like*
- > *an incredible option to not provide ASP users.*

I keep having flashbacks to "You are already logged into another workstation" messages from Novell clients after your computer blue screens. I don't know why, but I am. It's really foolish to build something into a product that can often give false positives. What you're suggesting would have that potential.

--

Ray at home  
Microsoft ASP MVP