ColdFusion Security Hotfix and Big Forms

Posted At : March 27, 2012 2:41 PM | Posted By : Cutter
Related Categories: ColdFusion, Adobe

The other day, Adobe released **a new Security Hotfix** for it's **ColdFusion** server. There were a number of things addressed in the hotfix, to help protect against Denial of Service attack using a hash algorithm collision. (My wife would say I sound like Charlie Brown's teacher right about now.) Ok, the important thing is you need to update your server.

Now for the fun part. We loaded the fix to our testing servers to run our app around the block prior to pushing this up to production. And, it's a good thing we did. We're preparing for a large deployment, and testing is pretty heavy right now. First thing in is that a form would no longer submit, throwing a 500 error every time. I didn't *show* me a 500 error, just a blank page. I had to look at Firebug to see the error code response. Now, if you've ever encountered a 500 error from the server then you know they don't typically tell you much. I reproduced the error locally and then went looking through the log files on the server.

In a multi-server configuration there are two core areas to look at log files. The first are the basic JRun logs. On a Windows systems, these files are located in the *C:\JRun4\logs* folder. Here you will typically find a *{instance}-out.log* file, and a *{instance}-event.log* file, for each ColdFusion instance you have configured. Right out of the gate you have *admin-event* and *admin-out* logs for the JRun administrator, and *cfusion-event* and *cfusion-out* for the default ColdFusion instance. I checked both files for my instance, and saw there weren't any items to tell me about the 500 error, so I then went looking at the ColdFusion logs.

Each ColdFusion instance has it's own set of log files, that you can see in the logs viewer in the ColdFusion Administrator. That said, the ColdFusion Administrator is not really the best place to go through these files, especially when you're really having issues. At this point, you just want to open them yourself. First, you have to find them. You do this through your instance, *C:\JRun4\servers\{instance}\cfusion.ear\cfusion.war\WEB-INF\cfusion\logs*. You'll probably find multiple log files here, from the *application* and *eventgateway* and *mail* logs, to individual logs from *cflog* calls.

Our issue, with submitting our form, was answered by the JRun *-event* logs, which gave me a few error messages saying something like this:

```
03/27 07:52:00 error ROOT CAUSE:
coldfusion.filter.FormScope$PostParametersLimitExceededException: POST parameters exceeds the maximum limit specified in the server.
 at coldfusion.filter.FormScope.parseQueryString(FormScope.java:397)
 at coldfusion.filter.FormScope.parsePostData(FormScope.java:346)
 at coldfusion.filter.FormScope.fillForm(FormScope.java:296)
 at coldfusion.filter.FusionContext.SymTab_initForRequest(FusionContext.java:377)
 at coldfusion.filter.GlobalsFilter.invoke(GlobalsFilter.java:33)
 at coldfusion.filter.DatasourceFilter.invoke(DatasourceFilter.java:22)
 at coldfusion.filter.CachingFilter.invoke(CachingFilter.java:62)
 at coldfusion.filter.RequestThrottleFilter.invoke(RequestThrottleFilter.java:126)
 at coldfusion.CfmServlet.service(CfmServlet.java:200)
 at coldfusion.bootstrap.BootstrapServlet.service(BootstrapServlet.java:89)
 at jrun.servlet.FilterChain.doFilter(FilterChain.java:86)
 at coldfusion.monitor.event.MonitoringServletFilter.doFilter(MonitoringServletFilter.java:42)
 at coldfusion.bootstrap.BootstrapFilter.doFilter(BootstrapFilter.java:46)
 at jrun.servlet.FilterChain.doFilter(FilterChain.java:94)
 at jrun.servlet.FilterChain.service(FilterChain.java:101)
 at jrun.servlet.ServletInvoker.invoke(ServletInvoker.java:106)
 at jrun.servlet.JRunInvokerChain.invokeNext(JRunInvokerChain.java:42)
 at jrun.servlet.JRunRequestDispatcher.invoke(JRunRequestDispatcher.java:286)
 at jrun.servlet.ServletEngineService.dispatch(ServletEngineService.java:543)
 at jrun.servlet.jrpp.JRunProxyService.invokeRunnable(JRunProxyService.java:203)
 at jrunx.scheduler.ThreadPool$ThreadThrottle.invokeRunnable(ThreadPool.java:428)
 at jrunx.scheduler.WorkerThread.run(WorkerThread.java:66)
```

"Dude! What is that!?!" Well, luckily I had installed my security hotfix, locally, just that morning, so I remember reading over the instructions. In it's notes it had stated the following:

4. Customers who want to change postParameterLimit, go to {ColdFusion-Home}/lib for Server installation or {ColdFusion-Home}/WEB-INF/cfusion/lib for Multiserver or J2EE installation. Open file neo-runtime.xml, after line

   ```
   "<var name='postSizeLimit'><number>100.0</number></var>"
   ```

   add the below line and you can change 100 with desired number.

   ```
   "<var name='postParametersLimit'><number>100.0</number></var>"
   ```

Just a heads up, that *neo-runtime.xml* file is minified, so you'll want to *Find* "postSizeLimit" to get that statement in the right place. We tried that *postParametersLimit* value (100) and found that our form had more than that (many were hidden, but that's another post all together), so we adjusted the number to 200. After restarting the instance again, we tested the form once more with complete success.

Hopefully this will help someone else avoid this issue. It's important to remember that Adobe does try to document these types of situations with hotfixes, so when you run into issues they should be your first source of troubleshooting information.