Email Injection Attack Part II - More Information

Posted At : September 19, 2005 5:29 PM | Posted By : Mark Kruger
Related Categories: Security

This blog is a follow up to a previous post, on the **Email Injection Attack** exploit and its occurrence on CF servers. Several questions and comments that were added indicates to me that I wasn't clear enough in describing what I believe is actually occurring. Let me see if I can shed some additional light on the subject.

Let's say you have a contact us form that is sent to "joe@marzipan.com". The email message is set up to be "from" the person making the contact so that joe can easily click reply and respond. The user also has control over a part of the subject and the message. The code would look like this:

```
<Cfmail from="#form.from#" to="joe@marzipan.com" subject="Email From contact form:
#form.subj#">
      #form.message#
      </CFMAIL>
```

Simple enough right?

Suddenly Joe (your client) calls you and says, "I'm getting spammed by fishy messages!" When you look at the messages in question you see the following.

- The *from address* is *someRandomString*@marzipan.com - the same as the clients domain.
- The *Message* seems to consist of the same address.
- There are 3 or 4 right in a row with similar characteristics.
- There may be a bounced message and it may show an attempted header in the body that says "bcc: someemail@somedomain".

Well now, that certainly does seem suspicious. Is it spam? Well, if by spam you mean sending gobs of unsolicited mass email, then no, it's not spam. Technically it's a probing attack - most likely by a bot.

The bot finds a contact page and hit's it 3 or 4 times with several known methods of inserting headers into email messages. For example, it submits as the *from* address

```
somename@yourdomain.com + chr(10) + bcc: somehiddenEmailAddress@aol.com + chr(10)
```

If your scripting language *assembles* a mail message using a format - in other words if it concatenates headers and messages into a string representing a mail message - then this would result in a header entry like:

```
from: somerandomstring@yourdomain.com
bcc: somehiddenEmailAddress@aol.com
```

Being inserted into the message. The key is, the secret email address would then receive a message *from* your contact page. If the bad guy *does* receive such a message he has found out something very important and valuable. He knows that he can send email to any email address *from your domain*. You have in effect created an open relay for him to use on your domain. His next step is to work with your form to do the same thing with content by concatenating boundary and mail part commands. Should he succeed, he has a way to send arbitrary messages from your domain to whomever he likes - he can spam *through you*. The red flag that you have a problem is not receiving

3 or 4 messages at a time with bogus email addresses. The red-flag is if you suddenly see hundreds of bounces - or your traffic on your contact form goes through the roof.

The important thing to know is that those extra 3 or 4 messages are not spam - but merely probing looking for this vulnerabilty. CF is not vulnerable (as far as I know) to this problem because the CFMAIL tag controls the creation of headers and mail parts using the java mail classes.

So if you see those email messages from your domain explain what is going on to your site owners. Meanwhile there are a few things you can do to "fix" this problem. First, message formats dictate line breaks as the delimiter for headers in a mail message. If you use form parameters in "from" or "subject" you can check for the existence of line breaks and not allow the message if any are found. This is not a *perfect* solution because there may be browsers that have issues with line breaks - and you *may* end up shutting out those folks. Let me add, I don't know of any such problems, I'm just trying to cover my but (ha). Here's how you might do it:

```
<!--- check for format and line breaks
    --->
   <cfif isEmail(form.from) AND NOT find(chr(10),form.from) AND NOT find(chr(13),form.from)>
      <Cfmail from="#form.from#" to="joe@marzipan.com" subject="Email From contact form:
#form.subj#">
      #form.message#
      </CFMAIL>

   </cfif>
```

Of course you should already be checking for proper email format - so your regex might already be doing the trick.