From: David Cole
Sent: Friday, April 25, 1997 11:09 AM
To: Robert Welland; Philip Bogle; Chris Jones; Hadi Partovi; Tony Ciccone; Ben Slivka; Brad Chase; Yusuf Mehdi; Cornelius Willis; Tod Nielsen
Subject: RE: Another ActiveX security hole

First, there is no way we'd go through all the pain of releasing just this. there are things like the fileupload QFE, fixes for bugs in the IE 3.02 JavaVM, authenticode changes for date stamp and self extracting exes, and other changes.

Second, it takes the test team approximately 3 weeks of dedicated time to make sure we are releasing a solid full product. We spent 2 weeks on IE 3.02, and missing some key bugs. there are patches to create as well to save people the full download. We have to localize it in all appropriate languages, and test those. we have to prepare and test drops for AOL and other online service vendors, as well as a full IEAK release.'

Assume we could get all this put together by mid-May. That is when we need testing on IE4, else it slips.

I know it's fun to think about just these narrow little fixes that require virtually no code change, but that isn't the reality of the release machine and quality control efforts needed. I am the first one to engage this machine when I think the benefit is there, like I did for the 3.02 release, but I don't see it this time. Whenever the model relies on a 3rd party to do the right thing, we are exposed. The above is lot of effort to simply put a different finger in the same hole-in-the-dike.

-----Original Message-----
From: Robert Welland
Sent: Friday, April 25, 1997 10:54 AM
To: David Cole; Philip Bogle; Chris Jones; Hadi Partovi; Tony Ciccone; Robert Welland; Ben Slivka; Brad Chase; Yusuf Mehdi; Cornelius Willis; Tod Nielsen
Subject: FW: Another ActiveX security hole

The amount of development work is essentially zero (see below). Of course, this does not make an IE3.x release trivial. Where did the "slip IE4 at least a few weeks" estimate come from?

Bob Welland

-----Original Message-----
From: **Philip Bogle**
Sent: Friday, April 25, 1997 10:49 AM
To: Robert Welland
Subject: RE: Another ActiveX security hole

Trivial... The "MakeSafe" tests are encapsulated in a single function; removing the registry check involves commenting out a few lines.

-phil

-----Original Message-----
From: Robert Welland
Sent: Friday, April 25, 1997 10:34 AM
To: Philip Bogle
Subject: RE: Another ActiveX security hole
Importance: High

How hard is it to remove the registry code?

Bob Welland

-----Original Message-----
From: **Tod Nielsen**
Sent: Friday, April 25, 1997 10:31 AM
To: David Cole; Philip Bogle; Chris Jones; Hadi Partovi
Cc: Tony Ciccone; Robert Welland; Ben Slivka; Brad Chase; Yusuf Mehdi; Cornelius Willis
Subject: RE: Another ActiveX security hole

We are working with all of the commercial control vendors that we work with to check their controls and mark them correctly. However, there will still be a lot of controls that folks create that will be marked incorrectly and potentially cause damage.

At the moment, we are not getting any more pressure on this issue from a PR perspective. Folks interpreted the Symantec incident as a Symantec issue, and not a general problem with controls. Assuming it doesn't

flare up into a huge issue, we could risk it and wait until IE4 to have a general mechanism to fix this. However, if it flares up, it could be the straw the breaks the camel's back and puts the final nail in the ActiveX coffin. My nightmare scenario would be for some pissy press person to take a sampling of 20 activex controls on the net, and discover that 18 of them are marked incorrectly and hence expose users to "major risk".

The safest approach would be to fix it now and proactively address the issue. However, even when we are proactive these days, we get slammed more than our competitors. So there is a risk that by being proactive, Sun kills us with this one anyway. So is it worth slipping IE4 and potentially still getting slammed on this issue anyway? Tough call. If we could do this without slipping IE4, I would say we should fix it now.

Bottom line, since I think doing this could slip IE4 by at least a few weeks, my recommendation would be to roll the dice and focus on getting IE 4 out the door. But we need to make sure we have some mechanism to address this in IE 4. We'll get all of the major control vendors to mark their stuff correctly now, but there is no way we will reach everyone. We are exposed on this one either way. We should all say a prayer over this one.....

- Tod

-----Original Message-----
From: David Cole
Sent: Friday, April 25, 1997 8:45 AM
To: Philip Bogle; Chris Jones; Hadi Partovi
Cc: Tony Ciccone; Robert Welland; Ben Slivka; Tod Nielsen
Subject: RE: Another ActiveX security hole

I am only guessing, but shutting down the registry based "safe for scripting" mechanism is sure to cause many incompatibilities between pages and controls. To fix this as an ICP, I think I would just whack together the IObjectSafety interface to allow my page to do it's thing again, including any unsafe interfaces that let me whack on the registry, get file names or whatever.

I agree we need a campaign to get controls cleaned up, but reving IE3 is a pretty extreme step to start the campaign. TodN, is there another way?

-----Original Message-----
From: Philip Bogle
Sent: Friday, April 25, 1997 8:26 AM
To: Chris Jones; Hadi Partovi; David Cole
Cc: Tony Ciccone; Robert Welland; Ben Slivka; Tod Nielsen
Subject: Re: Another ActiveX security hole

An IObjectSafety implementation is roughly a page of code. Using either the implementation provided by AFC 2.0, or a stand-alone implementation that I wrote, a developer can add the interface in ten minutes or less by simply inheriting from the appropriate implementation.
But we shouldn't give away the keys to someone who doesn't know how to drive-- we've got to make sure that people read the documentation and understand what it means to be "Safe for Scripting" or "Safe for Initialization from Untrusted Data". Too bad we can't convince the compiler to quiz the developer before building the code.
-phil

From: David Cole <davidcol@MICROSOFT.com <mailto:davidcol@MICROSOFT.com>>
To: Philip Bogle <philbo@microsoft.com <mailto:philbo@microsoft.com>>; Chris Jones <chrisjo@microsoft.com <mailto:chrisjo@microsoft.com>>; Hadi Partovi <hadip@microsoft.com <mailto:hadip@microsoft.com>>
Cc: Tony Ciccone < tonyci@MICROSOFT.com <<mailto:tonyci@MICROSOFT.com>> >; Robert Welland <robwell@microsoft.com <mailto:robwell@microsoft.com>>; Ben Slivka < bens@MICROSOFT.com <<mailto:bens@MICROSOFT.com>> >; Tod Nielsen <todn@MICROSOFT.com <mailto:todn@MICROSOFT.com>>
Date: Thursday, April 24, 1997 10:13 PM
Subject: RE: Another ActiveX security hole

How much dev work is typically involved in implementing IObjectSafety?

-----Original Message-----
From: Philip Bogle
Sent: Thursday, April 24, 1997 6:23 PM

To: David Cole; Chris Jones; Hadi Partovi
Cc: Tony Ciccone; Robert Welland; Ben Slivka
Subject: Another ActiveX security hole
Importance: High

We desperately need to deliver a patch to IE3 that plugs the rash of security holes opened by controls incorrectly marked "Safe for Scripting". MSN was just one example; there are several others. Fortunately, there's a fairly easy fix, which I urge you to take.

I realize that there are currently no plans for an IE3.10, but I urge you to reconsider.

To emphasize this point, I went yesterday to www.activex.com and looked at their *featured* control, called EasyMail. There (for any hacker to read) was an explanation of the control's features, which included the ability to address messages, attach any file on the users machine, and send the message off, all without any user intervention.

For example, if an administrator on an NT machine browsed to a hostile page, that page could mail the password file to the hostile user as an attachment, to be cracked remotely using any of a number of tools availabe on the web.

In a short time, I created a small demo of this control, which sends me a copy of your config.sys and autoexec.bat when you visit the page. http://philbo/mailhack.htm

This example is an addition to my earlier one, which allows a hostile page to upload and execute arbitrary code on the browser's machine
: << Message: Another security hole, using well-intentioned third party control >>

We've got to do something about this for IE3, or we risk major embarassment. It took no cleverness at all for me to realize the control was dangerous and to create a page that exploits it; it's only a short matter of time before someone else does as well. Furthermore, our accountability story falls totally to pieces. It's the page that's hostile, not the control (which is simply incompetent), so Authenticode is no use whatsoever.

The fix is fairly simple, but will require some coordination with control vendors. All of the incorrectly marked controls are marked using the registry, not the IObjectSafety interface. This isn't coincidence; the registry marking is trivial and can be done without reading the documentation; whereas the interface requires that you actually read and think about the docs and what "Safe for Scripting" actually means.

So, we should get rid of the code that looks in the registry to see if controls are safe for scripting and require IObjectSafety. We should tell control vendors that they need to change their implementation if they were relying on the registry entries. To reduce the impact of this change, we can special case the CLSIDs for certain very popular controls that don't implement IObjectSafety, if those controls are in fact safe.

Ideally, we would also have some sort of review process for control authors where we would review the design of the control and make sure it is in fact safe for scripting. We wouldn't be guarding against fraud, but we would guard against basic incompetence, which is a useful thing in itself.

I really think we need to take a stand and take some part in the certication of safe ActiveX controls... as I've often said, if Microsoft can't decide which ActiveX controls are safe, who can?

-phil