From: David Cole
Sent: Friday, April 25, 1997 10 07 AM
To: Robert Welland, Philip Bogle, Chris Jones, Hadi Partovi, Michael Toutonghi
Cc: Tony Ciccone, Ben Slivka, Tod Nielsen
Subject: RE Another ActiveX security hole

Nobody is disputing how bad this looks. Nobody is disputing that we should be proactive. The question is whether we launch an evangelism effort and do a release of IE, or just launch an evangelism effort.

I personally don't see the technical benefit of doing an IE release. I do see some evangelism benefit of doing an IE release. TodN needs to tell me if he can achieve 90% of the same benefits without a full IE release before IE4. We can get people to clean up using the upcoming IE4 releases as an excuse since we'll be super restictive there.

Again, reving IE3 is an extreme step since that will blow our IE4 schedules again. Releases are not free.

-----Original Message-----
From. Robert Welland
Sent Friday, April 25, 1997 9 51 AM
To David Cole, Philip Bogle, Chris Jones Hadi Partovi, Michael Toutonghi
Cc Tony Ciccone, Ben Slivka, Tod Nielsen
Subject. RE Another ActiveX security hole

I do not think we can avoid getting slammed on this issue Two reasons for being proactive

- It completely undermines Authenticode, a control from a reputable vendor can be exploited to do harm How can I, as a customer, believe Authenticode in the presence of this security hole?
- The registry fix Philip suggests will disable all of the bad controls Philip has found so far; this will significantly reduce the possibility that someone will exploit this security hole

I can only imagine the kind of bad press that this security hole will cause It is FAR worse then anything Sun has screamed about A few ugly scenarios

- The NT security hackers use the e-mail hole to get at sensitive admin files, this undermines our claims that the NT password hack is implausible
- The German hackers can probably compose a page that extracts Intuit financial information without reverting to un-trusted ActiveX controls

I don't understand what our alternatives are going to be Michael Toutonghi will be altering the "safe-for-scripting" model when he introduces the new security model for Java This will greatly improve our security story - and I think we will want to start outdating the old "safe-for-scripting" model Since we are going to break the old model why don't we phase it out and get the benefit of being proactive? If this can be done in the context of an MS security initiative then we might get some really good press instead

Bob Welland

-----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 roving 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>
To: Philip Bogle <philbo@microsoft com>, Chris Jones <chrisjo@microsoft com>, Hadi Partovi <hadip@microsoft com>
Cc: Tony Ciccone < tonyci@MICROSOFT com <mailto tonyci@MICROSOFT com> >, Robert Welland <robwell@microsoft com>, Ben Slivka < bens@MICROSOFT com <mailto bens@MICROSOFT com> >, Tod Nielsen <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 available 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