From: Tod Nielsen
Sent: Wednesday, April 23, 1997 3:49 PM
To: Michael Toutonghi
Cc: Charles Fitzgerald
Subject: RE: Safe Win32 Concerns

I like this, and generally agree with your points.

However, zones aren't exactly what I thought they were going to be. Zones still require a trust based decision by the user or the administrator. They are good and should help some, but they are still inherently dependent on trust. Whatever happened to the notion of having the NT guys come up with a "secure region" that NT users could use to run anything they want in, and be sure that code executing in that "region" could write to the file system, or do malicious things. In other words, I would log on to my machine with my account and get my normal permissions, but IE could be configured to only have guest permissions that had restricted access to files etc.

My understanding is that this would be an easier thing to do than try the magic "Safe win3" apis, and it would have the same "calming" effect on users.

Regarding your points on AFC and our APIs, the fundamental question that we need billg and paulma to decide is whether we are going to pursue a "class library/vm" api story, or a "tied to the os natively" one. I realize this oversimplifies the issue, but over half of the consipation we are suffering is due to the fact that we can not agree on this. You guys are clear, and push to have us do AFC everywhere and make our VM kick ass and support any language etc. Jim, Billg, and paulma to some extent hate this approach, and beat the shit out of me as we try and talk to developers about this.

We need to get an aswer to these questions asap, because it is crystal clear that we are going to lose if we sit on the sidelines and give a half hearted effort to both approaches. We need to decide on a strategy, and bet the farm to make it win. If there is anything I can do to help get this decision made, please let me know.

- Tod

—Original Message—
From: Michael Toutonghi
Sent: Tuesday, April 22, 1997 11:46 PM
To: Tod Nielsen
Cc: Charles Fitzgerald
Subject: FW: Safe Win32 Concerns

original mail is a few messages back

—Original Message—
From: Michael Toutonghi
Sent: Tuesday, April 22, 1997 11:18 PM
To: Jim Allchin (Exchange)
Cc: Paul Maritz; Paul Gross; Moshe Dunie; Deborah Black; Ben Slivka; Bob Muglia (Exchange); Brad Silverberg; John Ludwig
Subject: RE: Safe Win32 Concerns

It's true that Zones should integrate with the proxy server/firewall. Until that's done, SSL, PICs certificates, or signed HTML will be the only secure site validation. I believe that IE 4 will only support SSL. People are asking for IP mask/subnet support, and we should provide it, but it cannot be truly secure without firewall integration.

My point about ActiveX is that we all (including me) make statements about ActiveX controls being insecure, and we let the press get away with referring to ActiveX controls as lacking a sandbox. In reality, our upcoming VM release supports sandboxed ActiveX controls written in Java. (press mode) ActiveX is a language neutral component model which supports both sandboxed and non-sandboxed languages. Unfortunately, we all seem to know that, but we continue to refer to Java controls as applets (Sun's term) and Native controls as ActiveX.

My suggestion about AFC assumes a few things.

MSS 0003550
CONFIDENTIAL

ATTORNEYS ONLY

GOVERNMENT
EXHIBIT
1334

- Each group exposing platform specific Windows APIs to Java will make those APIs as Java/VM friendly as

possible. This API exposure will complete our strategy of providing a very slippery slope (leading to Windows) to developers.

- We expand both our platform dependent and platform independent Java APIs. We make APIs platform independent only as necessary to compete with Sun's cross platform offering and keep developers using Microsoft APIs. We remain on the treadmill only until our platform specific API exposure/innovation can outpace Sun's platform independent work enough to gain critical mass. If our platform independent APIs really compete with our own Windows APIs, it's only because Sun's did in the first place.

- If/when Sun's cross platform APIs truly compete with our Win32 APIs, we will either have had an alternative for developers or we could lose them.

- We must continue to truly innovate on Windows.

I see this strategy as a 1-2 year plan topped by the best Intel P7 JIT (Java/VB/Cobol/Ada/Objective C? by that time) compiler which would be available concurrently with the release of the P7. If there was a question about the superiority of the Windows platform, we will have answered it by then.

Thanks,

Mike

—Original Message—
From: Jim Allchin (Exchange)
Sent: Tuesday, April 22, 1997 7:58 PM
To: Ben Slivka; Bob Muglia (Exchange); Brad Silverberg; John Ludwig; Brad Silverberg; Michael Toutonghi
Cc: Paul Maritz; Paul Gross; Moshe Dunie; Deborah Black
Subject: RE: Safe Win32 Concerns

I actually don't know if everyone is so far apart on this issue.... A few comments ...

1. Active X has never been tied to a language. Whether the control is interpretive or native has never mattered. I was on a panel against Marc from Netscape last summer and I explicitly hammered him on this point. We should never be marketing Active X as requiring native code. If we are, it is a mistake.

2. Zones must be secure. I have not seen a plan to make them so. There is work required to ensure that the IP addresses are not spoofed. Security work is required before we will get away with the zones concept.

3. I agree win32/win64/etc. should be callable from Java. This seems obvious to me.

4. The comment about AFC concerns me since I think you are saying that whatever functionality is added by SUN you will add in a compatible way. They have you on a treadmill. I don't understand how this is a winning course. I would explicitly be different -- *just to be different.* Further, I believe the highest priority is to create and expose NEW Windows APIs (not clone) -- absolutely callable from Java. I would add more to Windows and less cross platform. This is clearly a hard tradeoff that will be made each day, but I believe it is not a question of how much resource is applied between Windows and cross platform. I believe it is a philosophy.

5. I agree VB should be unified into the model.

MSS 0003551
CONFIDENTIAL

6. I agree with the partners message.

ATTORNEYS ONLY

—

So, basically, I think I sort of agree with the above (except for the AFC treadmill issue). However, this plan leaves C++ users out in the cold -- except within specific zones. And I wonder why anyone would bother if Java worked in all the cases. Is that what we want to do? We will drive more people to Java faster. Given this, without something to pollute Java more to Windows (show new cool features that are only in Windows) we expose ourselves to more portable code on other platforms -- especially if the Pure Java logo push wins in any way. Either we need to pollute to Windows or we need more other languages to offer viable alternatives to Java (or both). I would feel so much better if I felt we weren't on such a middleware clone SUN strategy. This is where I are apart.

I must admit that one thing that is clear from this is that what is being said is that VMs are safe and the OS isn't for object execution. Given the failure of tagged architectures, the 432, etc. that may be the right model for

now. I need to ponder this over the next day to think if there is something we can do in the OS with memory management that might help this.

jim

From: Bob Muglia (Exchange)
Sent: Tuesday, April 22, 1997 6:46 AM
To: Ben Slivka
Cc: Paul Maritz; Jim Allchin (Exchange); Paul Gross
Subject: RE: Safe Win32 Concerns

I agree with Mike.

bob

---Original Message---
From: Ben Slivka
Sent: Tuesday, April 22, 1997 12:15 AM
To: Bob Muglia (Exchange)
Subject: FW: Safe Win32 Concerns

Your thoughts?

---Original Message---
From: Michael Toutonghi
Sent: Sunday, April 20, 1997 11:03 PM
To: Ben Slivka; John Ludwig; Brad Silverberg
Subject: Safe Win32 Concerns

I feel that I need to say this until I'm proven wrong, it makes a difference, or my stock options lose their value.

I'm very concerned about the path we're closing on WRT Safe Win32. First, I want to state our goals as I understand them. Then I'd like to suggest a more viable, efficient alternative. As I understand it, we want to achieve the following with the Safe Win32 effort.

- Provide a language neutral Internet development platform (keep people from moving to Java)
- End the debate over Java class libs & ActiveX/Win32 safety
- Preserve the Win32 API franchise as the Windows umbilical cord

I haven't heard other goals, so I'll start with these. First, I believe that we have other alternatives which have been prematurely dismissed as doomed to failure. I also believe that the Safe Win32 effort will most likely result in the opposite of the desired effect and could eventually seal the coffin of our Win32 franchise.

- Stated simply, I believe that creating this API will prove much more difficult than original estimates, even if they are conservative. When finished, our first release could easily have serious security holes as most any reasonably complex system has in its first release. Of course, by that time, competitive Java class libs will have most security holes fixed and will have a large and growing development community. People will continue, probably accelerate their move to the Java runtime. If Java is the only well supported language targeting the runtime, people will be moving to Java in droves.

- If we release an inferior API set that is safe and try to hold it up against Java class libs, we will not just end the debate over ActiveX/Win32 safety, we will begin and end the debate over Java class libs and Win32 functionality. **Safe Win32 will serve as a proxy to functional Win32 for this argument.** I predict it will not end in our favor. I see this as taking the worst of Win32 and pitting it against the best of Java class libs.

- Finally, after it is evident that we are trying to sell an API which is inferior to Java's class libs for the job of Internet development, developers will likely (and incorrectly) conclude that Java is superior to Win32. Game over.

MSS 0003552
CONFIDENTIAL

We do not have to take this course of action. I believe we have a much better strategic choice which will

ATTORNEYS ONLY

ensure our success/leadership if we execute properly. I recommend the following:

- Before it's too late, separate ActiveX from native code. If we lose ActiveX, we lose a huge amount of credibility. The market (and Microsoft in general) still equates ActiveX with native code, and this will destroy it for Internet/extranet development. Today we allow development of ActiveX controls in Java and we need to be committed to providing other languages (VB) for the Java runtime. Aside from my talks, we have NEVER articulated as a company that Java is a good language with which to develop ActiveX controls.

- Put together a white paper on Zones for the extra net and start pounding the table about it. **We need to drive and push this technology as the recognized leaders before someone else does.** Native code is perfectly suitable inside highly trusted zones.

- Make it a companywide mandate to expose all new APIs to Java. We've taken the first step in the VM and can now call Win32 APIs from Java, but our group can't do the effort which really belongs with each platform group of making all APIs easy. This will have the effect of making developers who can count on Zones take advantage of many APIs which are not yet available in Java. We need to start this immediately. The longer we wait, the more Java APIs we have to compete with.

- Keep the developers which will inevitably defect to Java in our camp by providing two sets of class libs. First, we make sure that AFC is cross platform, better than JFC and Sun's class libs, but not so much better that it competes with our platform specific APIs. If we have to make it compete with our Win32 APIs in order to keep it competitive, then Sun's APIs severely threatens us anyhow. We should be able to keep these cross platform features competitive with Sun's through appropriate resource allocation.

- Bring other languages to the Java runtime ASAP. **As soon as we provide any language alternatives to Java which target the runtime, Java loses all status as a phenomenon and becomes just another language.** VB is a must, and I don't give a shit about 50% perf on some unlikely scenarios, Microsoft needs the VB team to target the VM asap.

- Work agressively with partners to take the mindshare away from Sun. We should buy controlling interest in Metrowerks or stop Sun from buying them. We should work with Fujitsu to have them target our VM with Cobol (another reason Java is "just another language"). We should try to get Apple/Next to support the VM with Objective C, and we should make sure we succeed in our negotiations with HP.

I'm not saying that we shouldn't work on Safe Win32, but I don't think we should see it as a winning API strategy. Maybe in 2 years time, we could have a functional API set for Internet development that rivals Java. Maybe by that time we will also have the best VM class libs for Java, VB, C++ subset, Cobol, and Smalltalk. Ideally, they'll be largely platform dependent on Windows and no one will care about Safe Win32.

Thanks,
Mike