From: Brian Valentine
Sent: Friday, August 09, 2002 9:21 AM
To: Mike Beckerman; Jim Allchin; Will Poole
Subject: FW: Sfp api and WM setup

Importance: High

I asked Lonny for comments… obviously he is fired up about this…I am not going to get in the middle of this – you guys should decide what's right/makes sense based on where you are at and go from there. Anytime there are exceptions, it's bad, and we do get ourselves into weird places like having to doc or take exception on doc'ing the apis…

---- Original Message ----
From: Lonny McMichael
Sent: Friday, August 09, 2002 7:39 AM
To: Brian Valentine; Jim Allchin
Subject: RE: Sfp api and WM setup
Importance: High

Yeah, I got some comments...

---- Original Message ----
From: Brian Valentine
Sent: Thursday, August 08, 2002 6:13 PM
To: Lonny McMichael
Subject: FW: Sfp api and WM setup

Fyi… comments?

---- Original Message ----
From: Mike Beckerman
Sent: Thursday, August 08, 2002 6:11 PM
To: Will Poole; Brian Valentine
Cc: Jim Allchin
Subject: RE: Sfp api and WM setup

Here's the situation.

1) The SfcFileException API was called by our WMP 7.0 player setup that shipped back in 2000, and again by our 7.1 player setup that shipped in 4/'01 and that is still available on the web today (and will remain so until we've RTM'd the Corona player.)

Privileged

Corona is irrelevant to that. **SP1** is irrelevant to that. Obviously given the security implications we're going to call it out as an exception.

Obviously those products are "water under the bridge", and nothing can be done about them now. But that is really irrelevant to the discussion of whether the WMP team continues to circumvent WFP in this manner. It should also be noted that neither of the aforementioned versions installs on Windows XP, thus if we changed the interface for XPSP1 and .NET Server, it wouldn't affect those products.

2) We are driving toward RTM – lockdown for our RC1 test pass is 8/19; 10 days from now. We use exception packages where possible, however, we will not be able to completely remove all uses of the API.

If they truly did use exception packages, there would be absolutely no reason for them to be by-passing WFP. I would seriously question whether they use exception packages anywhere. You guys should be aware that when the Exception Package process was original designed, it was envisioned that the WinSE team would be the gatekeepers of the signature, and that out-of-band components would have to get their catalogs signed by the WinSE team (thus allowing us to track and audit their submissions, making sure they're doing the right things, etc.). However, the WinSE team decided

Plaintiff's Exhibit
8585
Comes V. Microsoft

not to spend resources here, and simply granted signing authority to every team doing exception packages. Thus, the WMP team may have in fact created a catalog and signed it with the Exception Package certificate, but that does not an exception package make. (There's a separate interesting question of exactly *what* they're doing with any catalogs they've gotten signed, because even if they were installing the catalog directly (i.e., not via Exception Package mechanism), its presence at the time of file replacement would satisfy WFP's digital signature verification, thus their new files should've been left alone.

I think it's safe to say that the WMP team decided not to spend any resources on figuring out what they were doing wrong, and instead found this back-door and decided to do a quick-and-dirty hack and move on to more and more "cool features". Of course, serviceability, upgradability, and stability of the OS aren't cool features that WMP can market.

We've worked our butts off to try to catch any and all reboot scenarios from our setup and there's a very high degree of risk that abandoning this API will introduce new cases.

Don't buy this for a minute--this API has absolutely NOTHING to do with reboots. They keep trying to use this argument, and there's no truth to it. There's nothing inherent in the exception package mechanism that necessitates a reboot. If you have files that are in-use by someone, then by default we have to queue up the new files for delayed rename. If they want to avoid this, then they would need to ensure that those files aren't in-use prior to laying down new files, *irrespective of the method by which those files are copied.*

I suspect that the WMP team has simply renamed the existing in-use files to some other name (e.g., **foo.old**), then are copying their new file to its final name Of course, anyone who has the existing file in-use will continue to use the old version, and the mix of old and new may cause problems, depending on the nature of the binaries affected. (Thus, the user might experience weird instability that would necessitate a reboot anyway. Only now, instead of being explicitly told they needed to reboot, they just experience what they've come to expect from Microsoft--system inexplicably goes "weird", so it's time to reboot again.)

Having said the above, if it truly is safe to do this, then they're still covered, because INFs can specify an "immediate replacement" flag that performs this same action. Typically, it is assumed that a reboot will immediately follow anyway (setupapi provides this flag for components that load early on before delayed renames can occur, so that we're sure to get the right kernel, hal, etc.). They could, however, suppress this reboot prompt if, as they claim, this is safe to do.

To recap, the WMP guys always raise the reboot issue (Jim, I'm sure they think it'll get them a lot of mileage with you, since I know you've beat them up before about requiring reboots--and rightly so)., but they never back it up with substantive information about what exactly it is they've done, and why that prevents them from using exception packages. Instead, they refer vaguely to "working their butts off" and the "special-casing and other beautiful things" they've done.

Privileged

4) We have no plans to do any other player release until Longhorn.

So? All this means is that they can continue to ignore the issue.

5) If and whenever we have to do another standalone release package after Corona, we will no longer call that or any other undocumented API.

Given their past track record, I think the only thing that would ensure they stop doing this is to make it so they can't. I've been pushing to make sure that we fix WFP for Longhorn so that these "back-doors" go away. I've been talking to the filesystem guys about this, and this is integral to our "consistent driver install" (aka, "driver lock-down") story for Longhorn. (Jim, I think this would be a good topic to discuss in our upcoming review with you)

We *can* force these guys to do something now. If we change the WFP interface, then those guys will be forced to either (a) figure out the new "back-door" or (b) fix this right. It's distressing that we'd have to consider such a possibility, but this is an option. This is what BrianV referred to when he mentioned that the issue was time-critical for XPSP1.

the effort is more than just a simple code change. It's serious work for us to then find and fix new reboot scenarios if that's indeed possible. Our install matrix includes Win98SE, WinME, Win2K, and WinXP, and service pack variants Given that the API must be called out as an exception anyway we just don't have the luxury of time to make this change, work all of the test permutations (and likely slip RTM), all in the interest of purity.

I wonder how these guys rate on the "integrity" values the company is focusing on these days? They ignore this problem for as long as possible, then claim it's too late to fix it. This unfortunately isn't the first group who doesn't think about their component's robust installation and serviceability until the 11th hour, then claim it's too late and they'll "do better next time". Of course, when next time comes around, once again they've been too busy focusing on new features to worry about such mundane tasks as ensuring their component installs cleanly, can be serviced, upgraded, etc.

You guys now how hard we've been trying over here to try to get off of old stuff and get resources moved to Longhorn.

How can they claim that a product that hasn't even shipped yet is legacy (i e., "old stuff")???

We just wrapped our SP3 work, are still finishing testing of SP1, and we've got all this Corona work to wrap up. Looking at the complete picture, I don't believe that worrying about this API in Corona is the right business trade-off.

I thought part of the "business trade-off" meant not shipping software that screws customers. As you know, we just didn't come up with WFP and Exception Packages to give other teams busy-work. This mechanism is critical to ensuring that we can properly service the OS, do the right thing when upgrading, and in general, avoid DLL Hell. I do not see a strong interest from the WMP team in these aspects of what it means to ship a quality product.

That said, as always, if you make the business call to do this anyway then we'll execute as effectively as possible.

-Mike
---- Original Message ----
From: Will Poole
Sent: Tuesday, August 06, 2002 6:30 PM
To: Brian Valentine; Mike Beckerman
Cc: Jim Allchin
Subject: RE: Sfp api and WM setup

I was not aware and will look into this with Mike asap.

thanks

---- Original Message ----
From: Brian Valentine
Sent: Tuesday, August 06, 2002 6:18 PM
To: Jim Allchin; Will Poole
Subject: FW: Sfp api and WM setup
Importance: High

According to the base guys, the media player found their own hack around WFP and didn't call the exception process the right way, etc… so when we documented the called for the compliance decree, we had to take an exception on the way it done for security reasons. According to Lonny, the player could fix this the right way – but he said they are getting a lot of resistance from the player folks. Are you guys aware of this? We have to make some decisions this week on SP1 and how to handle this. So it's time critical. I think the right answer is that the player fixes itself to follow the rules.

---- Original Message ----
From: Lonny McMichael
Sent: Tuesday, August 06, 2002 6:14 PM
To: Brian Valentine
Cc: Patty Esack
Subject: FW: Sfp api and WM setup
Importance: High

Brian, here's one of the early threads regarding Windows Media Player's use of the **SfcFileException** back-door. The more recent thread was atty-client privileged, and I've requested that Sue Glueck (the LCA representative on that thread) forward the thread to you.

Thanks, Lonny

----- Original Message -----
From: Lonny McMichael
Sent: Tuesday, February 26, 2002 2:14 PM
To: Zach Robinson; Scott Harrison
Cc: Marian Trandafir; Bob Fruth; Brett Miller; Erik Odenborg; Jason Cobb; Jamie Hunter
Subject: RE: Sfp api and WM setup

Below...

----- Original Message -----
From: Zach Robinson
Sent: Wednesday, February 20, 2002 5:24 PM
To: Scott Harrison
Cc: Marian Trandafir; Bob Fruth; Lonny McMichael; Brett Miller
Subject: RE: Sfp api and WM setup

Hmm. Recalling this **fully** may be difficult, as it was in 1999 and I purge mail regularly. The little I have in my old SFP folder **written in 1999**:
===

* Doesn't seem to work on RC2, work-around is to delete catalog file. Same package works fine on RC3? Work around is to delete our catalog files.

* Doesn't seem to work on various builds. Work around is to tell test we only support IDW builds.
Above seem to reflect the fact that WFP was unstable in its early days--no surprise, and not germane to this discussion.

* Doesn't version check on file installs, just overwrites. This forces us to have version checking logic in the package host applications.
This is very much by-design. Basing copy decisions on a per-file version number simply does not work. The versioning should be done at the package (i.e., component) level, and once the decision is made that a given package should be installed, then all files associated therewith must be installed to ensure package integrity (and maintain environment in which said package was tested/verified, etc.). This is not an argument against using exception packages, it's an indication that you are installing your files presently under broken assumptions.

* Beyonds specs and FAQs, seems to be little dev support for this. Since it's kind of flakey right now, that's pretty critical to us not getting bogged down debugging what should be trivial issues.
This reflects the fact that exception packages were meant to be few and far between, and our (naïve) approach was that if we made it harder to do an exception package, then fewer groups would attempt to do so. Instead, we found that they plowed right on ahead and either (a) circumvented WFP altogether (as you've done) or (b) constructed a bogus exception package, got signing authority from WinSE team, and proceeded to screw us by distributing packages that we could neither administer nor upgrade.

* At this point it requires us to use setupapi.dll to install our files. This means error recovery and reboot state issues and non-admin issues are out of our control.
Please expand on this point. What do you mean by "error recovery"? If an error occurs during setupapi queue committal, then we rollback the entire queue, so that the resultant on-disk state is left unaltered.

Also, could you elaborate on what "reboot state issues" you encountered? When setupapi is dealing with a signed package, it will not request a reboot unless absolutely required (e.g., if the existing file is in-use, and we must copy a new one over). To deal with this, you could ensure that the file(s) you're replacing aren't in use prior to committing the file queue.

I also remember that JasonC and I spent time with a couple of guys from the WMP team (sorry, don't remember their names) to assist them in developing a better algorithm for upgrading CD-ROM class filter drivers such that reboots were avoided if at all possible. (This was a result of JimAll encountering a reboot request when installing WMP.) The last I heard, that work was never incorporated into any WMP update.

Finally, w.r.t. "non-admin issues", this is simple. Non-admins should not be able to replace global in-box components. Period. If you guys are trying to address that, you're going to run right up against the security wall (if you haven't already).

===

I believe that what was happening was that we found Exception Packages were not working reliably. We got Andrew Ritz to look into our package, nothing was amiss, I believe Kirt Debique pulled in some security guy to triple-check that the test cert / catalog were being installed correctly, and everything checked out there too. I had high pressure on me to get this working, and it simply wasn't.

As far as specific bugs, I think the issue was with regards to not calling SfcFileException for the files, so they were being replaced when they should not have been. I believe I followed this one up with Andrew as well (perhaps someone else?) and they assured me that should not be a problem, whereas I found that my own implementation callling SFE fixed the issue.

Thankfully enough there is no third option on the table: **we are not and will not be talking about documenting this**, as it wouldn't make any sense to do so.

What the discussion thus appears to be about is WTF we did this. Am I correct? I was told I had two goals:

1. Make this work
2. Don't reboot

#1 wasn't being met at the time, and as far as **#2**, we have special-casing and other beautiful things you can do when you implement your own INF installer that drastically minimizes reboots. I have been told that I will be shot if I cause a machine to reboot, so I don't want to do so.
I'd like to know what "beautiful things" you're doing that setupapi wasn't. Since setupapi make all attempts at avoiding reboots, I'm inclined to believe that "beautiful" may equate to "slimy hacks", but I'll reserve judgement until I see your response.

These are my recollections offhand. If there are further issues/questions, perhaps we would be better suited to meet so we can have Q/A rather than the drawn-out exchanges of ... Exchange mail.

> ---- Original Message ----
> **From:** Scott Harrison
> **Sent:** Wednesday, February 20, 2002 4:56 PM
> **To:** Zach Robinson
> **Cc:** Marian Trandafir; Bob Fruth; Lonny McMichael; Brett Miller
> **Subject:** Sfp api and WM setup
>
> Zach can you describe the bugs we hit with the existing sfp
> implementation that prompted us to use the SFC dll api directly.
>
> I know the lack of file versioning is one issue are there others?
>
>
>
> As background for those not in the loop the current plans of
> the wm team are
>
> 1) ask for and get approval for WM setup to use this
> undocumented sfp api since it is a Windows Security API (we
> do this with drm for example)
>
> 2) change code to not use undocumented security / wfp API if
> exception is not granted. (unknown what the work is involved
> to do this)
>
> Documenting the SFP API is <u>NOT</u> part of this plan and is <u>NOT</u>
> acceptable to anyone involved here.