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 NOT part of this plan and is NOT
> acceptable to anyone involved here.