Re: A 6% fix from Microsoft Security Bulletin MS03−040 − 828750

Source: http://www.derkeiler.com/Newsgroups/microsoft.public.win2000.security/2003−10/1322.html

---

From: cquirke (MVP Win9x) (*name.goes.here_at_nospam.iafrica.com*)
Date: 10/11/03

```
Date: Sat, 11 Oct 2003 16:49:20 +0200
```

On Fri, 10 Oct 2003 15:59:16 −0700, "Me2" <nospam@nospam.com> wrote:

>CQuirke, you're a smart cookie! Bravo!

<blush> :−)

>Thank you for speaking out and illuminating the risks inherent in widening
>the surface area of software.

Surface area − that's a great analogy!

> have been kicking and screaming for years about this kind of
> stuff and have implemented many schemes to reduce the
> possibility that users will shoot themselves in the foot
> (in a corporate environment).

My beat is the opposite; "self−aministering" home and small business users, where one would like to restrict as little as possible but make it easier to practice "safe hex".

> Now I see many administrators who don't even see a problem.
> Until a black had rubs their nose in it...

What I like to do is generalize up from specifics to see if there's a theory−level lesson to be learned − then apply that theory to new situations and predict problems before they occur.

That's what theory is for, in this sense; to predict the bounds of the possible... many things that one would like to believe are impossible are merely difficult, or not even that; simply an opportunity no−one's grabbed yet.

>"cquirke (MVP Win9x)" <name.goes.here@nospam.iafrica.com> wrote in message

>> The above [see below]

Great directions, but context oversnip :−)

>> is the wider context in which to assess the answer to "if
>> the vendor knows of a defect that's being exploited In The Wild,
>> should users be informed and advised how to protect themselves?"

>I'm glad someone can see though the technology out to the real world.

When I see something that bugs me, I ask myself; is it due to an oversight, or by design? Part of assessing that is; what does the vendor stand to gain/lose?

>From MS's perspective, the wildcard in the pack isn't some new competitor running off with the market − that can be spotted some ways off. Instead, it's something that kicks over the legal antheap and attracts public as well as legal attention.

Even a reluctant, corporation−sympathetic government may have to act if the public is watching and prodding them to do something, and there comes a point when the impact on big business in general may cause them to Brutus in order to cut their losses.

That's why I see prudent modular design as being as much in MS's interest as anyone else's, even if this misses opportunities to "embrace and extend" into other markets.

> Microsoft should hire you.

Hey, they can read me for free :−)

>I understand that if you buy an new computer today with Microsoft software,
>bring it home, plug it in, wham! − in seconds − your PC is infected − ,
>reboot, reboot, reboot, install, reboot, repair, reboot go to fixit shop,
>bring it back... Interestingly − I hear from some − the incompetent user is
>at fault?

There's a triangle of capital, worker and consumer that causes business to work best when these forces are balanced in power. This facilitates the selection pressure that is often mooted as the most efficient way to progress.

No choice, no selection; that's the problem as most of us see it.

No pressure, no selection; that's the far larger problem.

In the industrial revolution, a situation arose where a small number of employers made it impossible for a large number of individual workers to apply selection pressure − because a small number of players can act in unison as a cabal. Organisation of workers into trade unions redressed matters somewhat, and has brought social benefits beyond the obvious (i.e. better wages for workers). For

example, union pressure can highlight public safety issues.

In the information revolution, a situation is arising where a small number of vendors make it impossible for a large number of individual consumers to apply selection pressure.

Right now, the software industry has been left to write its own rules, and the result is a disaster for users:
- long−term patents, then product is destroyed and useage denied
- limited license rights
- sanction on self−repair ("reverse engineering" etc.)
- no vendor safety liability whatsoever
- EUL"A"s that are drawn up unilaterally by vendor
- EULA's hidden inside no−return−if−opened boxes
- moving goalposts for expectations of service
- forced ongoing relationship with vendor
- vendor reserves right to unilaterally poison−pill user's system
- user to obtain "product recall" replacement code at own cost
- user to obtain "product recall" replacement code at own risk
- in some cases, proprietary lock−in where data is concerned
- in some cases, version lock−in and lock−out to force upgrades

Organizing the users is difficult because the whole shebang is likely to get hi−jacked by the industry − whether it be pro−vendor shills or (more likely) an agregate of "everyone except Microsoft".

And believe me, being dragged back to the mainframe era (now disguised as "thin clients" renting time on application servers) will re−invent the need for a MS to promote "personal computing".

>> In the DOS days, what the frontier was well−defined, and
>> 99.99% of attacks were made at the SE level. In fact I don't
>> know of any attacks that breached the frontier design

>Yes, until 1988. Least we forget the Internet worm of 1988.

DOS business and consumer PCs were pretty well immune to that sort of problem, as they hadn't gained their WAN wings yet :−)

>Did Microsoft architecture teams just forget? Maybe.

I think there's deffo an attention−defecit disorder there, i.e. a short attention span. Things are designed in a certain way for particular reasons, then those reasons are forgotton (seing as the design prevented those problems from happening) and the clue that informed the original design gets tossed overboard for some trivial "shiny things!" sort of reason.

For example, consider the matter of WinME's automatic file system cleanup after a bad exit. The original reason why this facility was deigned was so that by the time Windows starts to write to the file

system, the file system would normally have been repaired so that its now safe to do so. That's why the Win9x versions that had the feature had always done this using real−mode Scandisk, because that could complete the job before Windows starts writing to disk.

But because WinME has been tinkered with to make at look as if it's less connected to "DOS" than earlier Win9x, it now runs this automatic Scandisk while Windows is loading (and writing to the file system).

It's like making jumbo jet tyres out of glass because it looks so kewl when the sun shines through them on take−off, forgetting why jumbo jets needed landing wheels in the first place.

>...understood the problem and wrote future software with this in
>mind. In the late 1990s the lesson was ignored by Microsoft.

Yes, there's been a rather unfortunate NIH problem in effect. MS likes to pretend that the areas they extend into had never existed before they came along, inventing everything from scratch.

I can understand the joy of sweeping with a new broom, but part of that joy is not having to make the same mistakes.

An old design may become unwieldy because of a lifetime of corrections necessitated by the "school of hard knocks"; designing afresh should take those lessons on board and result in a smoother design that works as well or better. Re−inventing the *original* design and then having blunder through all the same mistakes from scratch is duuumb; at best it will result in a scarred but stable version years down the track.

Mind you, some ideas are so dumb that none of the pre−existing solutions ever made those mistakes − so there's no old newsreels of fiery crash−and−burn to watch and learn *those* lessons :−)

> If 9/11 taught us anything − it taught us to expect the possible.

Lovesan penetration + CIH payload = economic metdown.

The equation is that simple... imagine the business impact of not just one tower full of offices, but most of everyone's infrastructure (no matter how widely distributed) not only knocked offline, but blown away at the hardware level (think soldered−on BIOSs with no fallback, think proprietary laptop motherboards, think old motherboards tyhat can't be bought off−the−peg anymore).

>The "infosphere" looks like it's beyond our control − maybe the "malware" on
>the Internet is just a natural extension of our (life's) basic instinct
>for "survival of the fittest", evolution − that kind of thing.

The infosphere is indeed a selection−pressure environment, just like the biosphere, with one difference; mutations (and indeed software in

general) are made, they do not as yet spontaneously arise.

I've explored those themes recently in alt.comp.virus; a google on "infosphere" and "cquirke" will prolly find them.

It's certainly complex enough to defy deteriminism, and become interesting as a result. In fact, that's what interests me the most.

>There is no absolute solution. Its a kind of natural war with the
>offensive, containment, defense and coexistence we all know about.

Just as you can model natural complexity in a computer, so you can look at the most complex biosphere survivors (and see how these differ from simple ones) and learn lessons.

For example; bacteria have no nucleus membrane, and the genetic material just lies around, not really organised into chromasomes. Anything can squirt in new code, and just about everything does; bacteriophages, conjugation with other bacteria that aren't even the same species, human researches wanting to trick E.coli into manufacturing harvestable human Insulin, etc.

Multicellular organisms have a clearly−defined way in which code is exchanged, so they still benefit from breadth of variation as do bacteria, but it blocks out "dropper attacks". Wrong species; don't bother to apply. Right species, but looks dodgy? Forget it! Food that is eaten is smashed down into fragments too tiny to pass code unchanged, and only then does it enter the system.

Humans also have enough on−board and retrospective intelligence to predict problems and fix them, but alas, the sealed nature of the cell and nucleus makes it hard to apply direct maintenance. You may know that certain protiens are part of your own body, but you can't stop an inappropriate Rheumatic Fever immune reaction from shredding your heart valves. A hermetically−sealed, DCMA−enshrined future Windows could become exactly that kind of death−trap.

>Containment can be achieved at various levels within a system
>(network of machines). I have network containment controls (bulkhead
>controls) but when it comes to an individual machine − Microsoft's software
>architecture is out of my control − only Microsoft will decide to make
>containment within the kernel and user levels easier with compartmentalized
>design. UI compartmentization − Microsoft has gone to lengths to blend this
>level into a mash of indistinguishable S#@%

I have a hunch MS may just re−invent the entire wheel, and use whatever by−then glaring deficiencies there will be with our current Windows as reasons to upgrade. Except this time, you may never ne allowed to leave the shop with your purchase, and will be charged rent

Two lessons to bear in mind, on such a strategy:
1) Every layer is only as good as one it rests on
2) No code is ever perfect
3) Don't even let the system override the owner's control

The question is, who is the owner? The corporate administrator? For consumer PCs, the user via a simplified UI − or MS as the default system adminstrator of the world? Or none of the above; instead, a global cabal of media pimps and cronies?

>Defensive tactics is something we (administrators and Microsoft) have more
>control over than any other in the short term.

Yep − and it's something that has to be designed in. Not in the form of "in passwords we trust"; in simply making hostile actions impossible. That agenda may be best served in the corporate world via NT's existing model, but in the consumer world, users are already familiar with one better suited to their needs; physical access.

*>If I want to protect *my data* (i.e. the "queen", nucleus or DNA, etc). How*
>do I do it? With layers of defense that the enemy needs to circumvent.
>Alas, Windows makes this very hard.

Windows is learning its ABCs. For example, it now knows that data is best located on a per−user basis, rather than dumped within each program's own code files. It attempts to select system scope while ignoring personal data scope; that's what SR tries to do.

What it still doesn't have a clue about is the concept of "suspect files" or risk. It gaily dumps anything IE downloads into "My Documents" and nests Messenger's "My Recieved Files" thetre as well − the concept of "data hygiene" has yet to glimmer.

Now I am not for a moment suggesting attempts to track the origen context of a file as it passes through the system − that's what IE's security zone attempts to do, for example. But just as I'd never consider IE's zones to be even sand−tight, it's still a useful net.

Such awareness only becomes counter−productive when you consider it to be air−tight enough to confer all sorts of ill−advised rights to those "inside". As long as treated humbly, a "suspect zone" is useful.

*>Like you, I find it exceeding hard to isolate and control *my data* from*
>Windows and installed programs. Even a program assembly is sliced up and
>stashed all over the place in Windows − some pieces in "program files",
>%windir%, system and/or system32, the registry (user and system) and/or some
>other odd place.

>What happened to storing one program in one directory with rights?

That sort od scope awareness would massively simplify maintenance, and that's against the interests of the software industry. Not because they want you to have problems, but because they don't want any scrape−off of thier programs to be as functional as a formal install.

*>What happened to storing *my data* in one directory with access controls?*
>The "user profile" directory is a travesty that tries to store a user's data
>in one spot − in a complex way.

The nesting logic is asinine, IMO − why the hell would you want unsolicited incoming malware and your entire .MP3, videos and picture collection nested within your data set? How is that going to fit on limited−capacity backup media? How safe is that going to be to restore, after a delayed malware payload that also clears forensics?

It's the right idea, but it's too difficult to relocate these around (TweakUI helps) and − fatally − you cannot *pre*load where these things will be located when new accounts are created.

> Some of the user data and settings are stored in hidden directories
> (desktop, shortcuts, etc), some is stashed in a slice of the "registry".

Unless using roaming profiles, the relevant registry part is located within the user account subtree AFAIK.

The trouble is, the safety scopes are too mixed up. For example, any shortcut collection is a malware dropper opportunity, with StartUp as the big prize, and the same applies to registry content. That's why you can't write−share the whole of "Documents and Settings", or back up the whole of this subtree with safe−to−restore assurance.

My own practice is to locate small data on a small volume on it's own, and automate a backup of this to another small volume as an unattended Task. Such backups can be pulled onto other LAN peers to create a multi−redundant "holographic" store that resists all but total infrastructure destruction, even if no−one ever inserts a backup medium. Temporal depth can be created using the same multi−shot cyclical backup logic that Win98's RB00?.cab uses.

Data too big to automate in this way is stored elsewhere, as are pictures, video, music and other collectables that, while you'd want to see again, are not as irreplaceable as your own data.

Suspect files (emaul attackments, downloads, peer−to−peer file sharing bins, etc.) go in their own subtree, under the muzzle of a battery of on−demand scanners that can be launched via a single QuickLaunch click

This works well, until some idiot creates new user accounts. Then everything duhfaults to being in C:, which slows down as those absurdly huge per−user IE web caches fill up.

> there is no standard − every version of Windows changes the scheme.

Yep. MS seens incapable of maintaining any sort of continuity there;
that attention defecit problem again?

Data may be in; "C:\My Documents", %WinDir%\Personal,
SomeProfilePath\AsAbove, Local Settings\some\path, who knows? Why not
bury things in the bowels of the OS subtree? Duh.

The worst was when Office 97 unilaterally imposed "My Documents" as a
new "system folder" on the OS (since when does an *application* have
that authority?) without any UI other than RegEdit to manage it.

Unforgiveable, and a good Exhibit A why splitting MS into OS and
application companies would be of some benefit.

> Moving a users data between one PC and another is a
> nightmare − users hate it and grumble every time − unless
> administrators "fix" the situation somewhat.

I mentioned attempted pan−directory scope awareness earlier on, in
connection with System Restore. FAST attempts to do the reverse;
fillet out everything that *isn't* "system" and present this to you as
a restorable backup. Gary Woodruff clued me into this, and he has a
good page on the topic somewhere Googleable.

Both SR and FAST are version 1.1 and 1.0 (respectively) attempts at a
"mission impossible" brief, which rears fears of vendor hubris.
AFAIK, both are equally closed "trust us" systems, in that I know you
can't pick−and−choose amoungst SR's stores (you can't even clean
malware there <g>) and this may apply to FAST too.

Their value may not be in whether they work or not, but that a scope
awareness may better inform decisions on where to put things.

However, MS doesn't share the UNIX view that the file system directory
structure should also define other scopes (such as user rights, etc.).

I'm in two minds about this; frankly, a single structure can turn out
to be a straight jacket (hence Lotus Improv and Excel pivot tables −
and look how easy to use they are!). Scopes vary and overlap, e.g.
 − access speed are file system risk exposure (which volume?)
 − per−user vs. system
 − data / large−data / system
 − safe / unsafe−infectable−risky
 − in−system material vs. stuff coming in from outside
 − shared / read−only−shared / not shared

One could conjure up an elaborate system to manage all of these
many−to−many relationships, but it's all sandcastles in the air if
anything goes wrong within the levels beneath (malware intrusion,

flaky hardware that sees everyting as raw sectors).

There has to be a way "under" this to maintain it, else you have a data death−trap on your hands... an elevators−only skyscraper with no windows or fire escape. Be careful what you wish for.

Who has the rights to "maintain" the system at that level? The owner, of course. In consumerland, that should be a matter of who has physical access; if you want a virtualized but nearly−as−safe model to facilitate corporate remote admin, pay up for the Pro version.

If OTOH this "ownership" is taken out of the hands of the person buying the product... well, maybe it's time to storm the Bastille.

>The next virus/worm/trojan that comes walking through MY door − through the
>firewall, past the antivirus (with the latest up−to−the−minute updates −−−
>some forget this fact), past the email scanners − like Trojan.QHosts did −
>And Microsoft says "sorry not our problem − see the AV vendor" − I'm going
>to shout bloody murder again

You can't tune into a cabled LAN unless you have a pre−existing presence there. You can't attack a subsystem through a port that doesn't exist. You can't hide under the system's own access protection if the system doesn't deny the owner full access to anything. You can't get your code auto−interpreted if the interpreter not only doesn't interpret script, but doesn't have "unchecked buffers" (hint; StrCopy is not your friend). You can't break through password protection to use a service that does not exist. There's no point in crossing user account rights and security zones if the functionality you seek does not exist.

There's a clue meme in there, looking for some frontals to infect :−)

>−−−−−−−−−−−−−−−−− −−−−− −−−− −−− −− − − − −
The rights you save may be your own
>−−−−−−−−−−−−−−−−− −−−−− −−−− −−− −− − − − −