

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

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.win2000.security/2003-10/1312.html>

From: Me2 (nospam_at_nospam.com)

Date: 10/11/03

Date: Fri, 10 Oct 2003 15:59:16 -0700

CQuirke, you're a smart cookie! Bravo!

Thank you for speaking out and illuminating the risks inherent in widening the surface area of software. I 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). But every year, more and more "flexibility" is build in and over the last few years I have give up fighting it. Now I see many administrators who don't even see a problem. Until a black had rubs their nose in it...

"cquirke (MVP Win9x)" <name.goes.here@nospam.iafrica.com> wrote in message news:sdfcovsfn3ri6gsvjhm52g6jqfeq3u9kaj@4ax.com...

> *The above [see below] 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. "Risk management" what a non-techie thing to think about! I re-read many of your posts and see that you are explaining the situation well – somewhat as I see things anyway. Microsoft should hire you. But then that M\$ sales thing might blunt your sharp points...

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?

> >*When was it that viruses started effecting Microsoft OSes and apps? DOS v3*

> >*or was it v1. When was that? *1984* – almost 20 years ago. They just*

> >*started noticing? I don't buy this.*

>

> *The nature of the problem changed – and the reason isn't simply the*

microsoft.public.win2000.security: Re: A 6% fix from Microsoft Security Bulletin MS03-040 – 828750

- > *"oh it's so difficult!" cop-out excuse (i.e. that modern code is so*
- > *complex, we should abandon expectations that it works out of the box).*
- >
- > *In the DOS days, what the user needed to know was this:*
- >
- > *1) Files ending in .exe, .com and .bat are programs*
- > *2) Don't run programs unless you trust them*
- > *3) Don't boot off untrusted diskettes*
- >
- > *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 as enumerated above – not one.*

Yes, until 1988. Least we forget the Internet worm of 1988. I worked with DEC systems at universities and I remember it well. Did Microsoft architecture teams just forget? Maybe. Likely, every time someone inside Microsoft would say that adding a feature had a risk associated with it – someone else would counter with the "it's good for sales" thing.

[See: <http://world.std.com/~fran/worm.html> for more info on the Internet worm.]

As a result of the 1988 Internet worm Unix vendors (95% of the Internet at the time) understood the problem and wrote future software with this in mind. In the late 1990s web expansion of the Internet – the 1988 lesson was ignored by Microsoft. This is partly why they have this state of affairs.

<snip>

- > *At the risk of sounding like W.C., you are still thinking in terms of*
- > *the "virus infects computer" model. AV intercede on file operations*
- > *and keep a PC clean – but they cannot do anything to clean the entire*
- > *infosphere (the concept is absurd) or block DoS effects – and av don't*
- > *do anything at all within the risk management field.*

I agree! Anyone who is relying on an AV vendor to *stop* the next Internet bug should have their head examined. The AV vendors are doing a good job of mopping up after the mess Microsoft has made. I for one do not want to get caught with my pants down. If 9/11 taught us anything – it taught us to expect the possible.

Anti-virus vendors ARE doing a good job of monitoring – something that Microsoft is apparently wholly lacking in.

Microsofts latest stop gap measuers "Microsoft smears lipstick on a pig" will only help to a small degree. [i.e. <http://www.microsoft.com/presspass/press/2003/oct03/10-09SecurityInvestmentsPR.asp>]

Here we go again: As broadcast on NTBUGTRAQ today (10/10/03):

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

> Dear bugtraq@securityfocus.com,
> There are few bad news on RPC DCOM vulnerability:
>
> 1. Universal exploit for MS03-039 exists in-the-wild,
> PINK FLOYD is again actual.
> 2. It was reported by exploit author (and confirmed),
> Windows XP SP1 with all security fixes installed still
> vulnerable to variant of the same bug. Windows 2000/2003
> was not tested. For a while only DoS exploit exists, but
> code execution is probably possible. Technical details are
> sent to Microsoft, waiting for confirmation.
> --
> <http://www.security.nnov.ru>
> ^_^
> { , . } ^
> +---oQQo-->{ ^ }<-----+ \
> / ZARAZA U 3APA3A }
> +-----o66o---+ /
> //
> You know my name – look up my number (The Beatles)

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.

What is the solution?

There is no absolute solution. Its a kind of natural war with the offensive, containment, defense and coexistence we all know about. Offensive tactics like law enforcement have a limited effect when the perp can hide around the world. 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#@% – well – you say it nicely:

> And the data/program distinction is well and truly hosed, to the point
> that extensions are hidden and "leaky" (i.e. even if you can see the
> .ext, it can no longer be relied upon). So much nicer for programs
> and shortcuts to have their own unique icons, even if it means there's
> no replacement way to tell what is a program and what isn't.

Defensive tactics is something we (administrators and Microsoft) have more control over than any other in the short term. We bolster defenses every day with firewalls, AV, patching, etc. This will work to a good degree – but we suffer when the enemy finds a weakness in the wall or adapts to the defenses at the gates.

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.

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?
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. Some of the user data and settings are stored in hidden directories (desktop, shortcuts, etc), some is stashed in a slice of the "registry". Some of it gets stored around the disk in various program directories – there is no standard – every version of Windows changes the scheme. 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.

<sorry, I am rambling, I'm going to quit this thread...>

* * * *

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 – even louder – like it will help...

Me out

"cquirke (MVP Win9x)" <name.goes.here@nospam.iafrica.com> wrote in message news:sdfcovsfn3ri6gsvjhm52g6jqfeq3u9kaj@4ax.com...

> On Thu, 9 Oct 2003 10:06:15 -0700, "Me2" <nospam@nospam.com> wrote:

> > "whoever" <nospam@nospam.invalid> wrote in message

> > > "Me2" <nospam@nospam.com> wrote in

>

> > > *RPC was vulnerable for *5 YEARS* before blaster came along. It only took 3*

> > > *weeks after Microsoft informed the world about the potential problem for*

> > > *an exploit to be released.*

>

> *You can see how "NDA logic" applies here.*

>

> *A key thing to look for when signing an NDA (Non-Disclosure Agreement)*

> *is whether it applies to information that is no longer private (for*

> *whatever reason). A "good" NDA will remove restrictions on*

- > *information that is already public, whereas one that applies to*
- > *information whether it is public or not amounts to a gagging order.*
- >
- > *For example, if I got you to sign an NDA that stated "anything you*
- > *hear during the following seminar is private and must not be repeated*
- > *in public", I could stand up and say "our product sucks" as my*
- > *introduction. You would now be gagged from denigrating our product,*
- > *because you heard it here, even if you didn't hear it here first.*
- >
- > *OTOH a "good" NDA would leave you free to say "their product sucks" –*
- > *you just wouldn't be allowed to say "their own CEO says their product*
- > *sucks" until someone else made that information public.*
- >
- >
- > *In that spirit, a publically unknown hole can be treated as a private*
- > *matter (and should scramble an urgent chase to fix – 3 years is a long*
- > *time not to do this for something as evil as that RPC hole, suggesting*
- > *it was privately unknown too).*
- >
- > *Once publically exploited, the cat is out of the bag and there is IMO*
- > *an obligation for the vendor to 'fess up.*
- >
- > *If they have a patch; good.*
- >
- > *If they don't have a patch, then tell us how to wall out that*
- > *functionality for safety – even if it means having to concede that the*
- > *design or coding of that functionality is so poor that we should have*
- > *second thoughts about ever using it again.*
- >
- > *If the functionality is so embedded that it can't be walled off, then*
- > *this is urgent product quality information that is crucial to rational*
- > *planning – once again, it's information that cannot be withheld.*
- >
- > *You can't trumpet market forces as an acceptable referee and then rig*
- > *the game. The law recognises that in all sorts of ways, such as*
- > *"insider trading", for example. Continuing to cover up a defect when*
- > *it is publically exploited and thus a very clear and present danger to*
- > *consumers crosses that line, and spreads beyond the security debate.*
- >
- >
- > *Better yet, make sure you never geat into that situation. You know*
- > *there will always be coding defects, so you have to forego the hubris*
- > *of thinking that old safety standards are fuddy-duddy stuff you can*
- > *prance around with impunity. Never eat anything bigger than your own*
- > *head; never code a monolithic system that is bigger than your ability*
- > *to maintain fine-grain code quality and know/test *exactly* how*
- > *everthing works in practice, for all possible permutations.*
- >
- > *For a long time, modular program design was a big buzzword. There*
- > *were good reasons for that, and those reasons haven't gone away.*
- >

> >When was it that viruses started effecting Microsoft OSES and apps? DOS
v3
> >or was it v1. When was that? *1984* – almost 20 years ago. They just
> >started noticing? I don't buy this.
>
> The nature of the problem changed – and the reason isn't simply the
> "oh it's so difficult!" cop-out excuse (i.e. that modern code is so
> complex, we should abandon expectations that it works out of the box).
>
> In the DOS days, what the user needed to know was this:
>
> 1) Files ending in .exe, .com and .bat are programs
> 2) Don't run programs unless you trust them
> 3) Don't boot off untrusted diskettes
>
> 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 as enumerated above – not one.
>
>
> Now it would have been possible to evolve today's complexity while
> maintaining a frontier that was as well-defined as above; perhaps with
> even more user friendliness than above, something like...
>
> 1) Files with red triangle icons are programs
> 2) Don't run programs unless you trust them
> 3) Don't boot off untrusted media
>
> If you could trust data to not act as programs, a whole slew of
> problems go away – web page attacks, document malware, no-click email
> attacks, auto-running CD attacks.
>
> If you continued a sense of frontier awareness within the boundaries
> of the network, you'd have fewer escalation risks to worry about.
> With no scripting inherent in "View As Web Page", every write-shared
> folder would no longer be a land-mine opportunity. With \AutoRun.inf
> processing for HD volumes disabled, and write-shared volume root need
> not be a landmine opportunity. With those dumb-ass "admin shares"
> disabled, we could follow good LAN sharing practice and never expose
> the startup axis or system code base, and wouldn't have to care about
> password discipline or efficacy in that context.
>
> As it is, our struggle slogan "an injury to one is an injury to all!"
> applies, and that's not in a *good* way :-)
>
>
> That's why the situation is spiralling out of control.
>
> The need for corporates to centrally-administer PCs was allowed to
> override the need for home users to retain the meaning of the word
> "home" (a physical location where safety can be assumed).

- >
- > *The need for web sites to manipulate users for marketing purposes was*
- > *allowed to override the user's safety needs, and once HTML was the web*
- > *standard, no-one had the clue to see why simply using this as-is as a*
- > *system-wide "rich text" standard (including email) was a Bad Idea.*
- >
- > *The need to spare CD-ROM vendors from having to explain how to "click*
- > *Start, Run, enter ?:\RUN where ? is your CD drive letter" left us with*
- > *auto-running CDs; can't see what it is until it's already run itself.*
- >
- > *And the data/program distinction is well and truly hosed, to the point*
- > *that extensions are hidden and "leaky" (i.e. even if you can see the*
- > *.ext, it can no longer be relied upon). So much nicer for programs*
- > *and shortcuts to have their own unique icons, even if it means there's*
- > *no replacement way to tell what is a program and what isn't.*
- >
- >
- > *The frontier is so fuzzy, that it's near impossible for the average*
- > *user to practice "safe hex". And there is so much "dancing with*
- > *wolves" going on that even if the design doesn't give malware a free*
- > *backstage pass, there's such a maze of little band-aids to shore up*
- > *the frontier that code defect opportunities will abound.*
- >
- > *Today's malware mainly exploits bad software design, i.e. the*
- > *opportunities presented by the users' inability to assess the full*
- > *risk their actions facilitate. I don't expect "reading message text",*
- > *"visiting a web site" and "reading a document" to be conferring*
- > *programming rights to those entities, but they do.*
- >
- > *Today, user and vendor share responsibility for malware outbreaks; the*
- > *user, for not practicing "safe hex", and the vendor, for undermining*
- > *the user's ability to practice "safe hex".*
- >
- > *Tomorrow's malware may focus mainly on code defects rather than simply*
- > *leverage poor software design. In this case, the user's*
- > *responsibility is completely bypassed, and the vendor's responsibility*
- > *is manifest. Unless we make special rules for the software industry*
- > *to let them off the hook ("oh, it's so difficult!" etc.), there is no*
- > *question who to blame where a product defect is the cause of the*
- > *problem and the user's ability to manage this is sidelined.*
- >
- > *The above 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?"*
- >
- >
- >
- >
- > >-----
- > *Drugs are usually safe. Inject? (Y/n)*
- > >-----