June 11, 2009

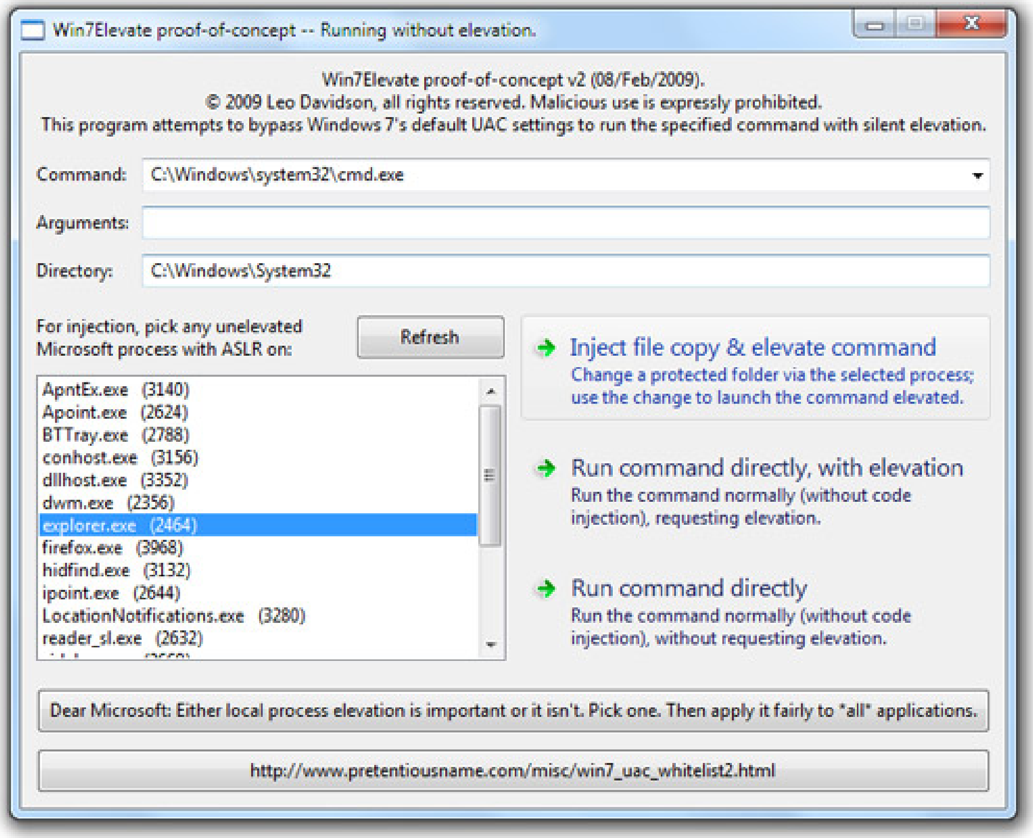
UAC in Windows 7 still broken, Microsoft won’t/can’t fix code-injection vulnerability

I admit, as a non-programmer, I have very little knowledge about the inner-workings of Windows. However, as an enthusiast, I thought I had a basic but firm understanding of what User Account Control is, how it works, and why it exists. That’s no longer true. After reading reading an article by *Windows-god* Mark Russinovich, “Inside Windows 7 User Account Control“, I’m bewildered by the changes to UAC in Windows 7.

At first, Mark provides this logical explanation for UAC elevation prompts.

> Elevation prompts also provide the benefit that they “notify” the user when software wants to make changes to the system, and it gives the user an opportunity to prevent it. For example, if a software package that the user doesn’t trust or want to allow to modify the system asks for administrative rights, they can decline the prompt.

Bearing this in mind, you’re probably familiar with the commotion raised months ago over a concern over how applications can silently turn off UAC prompts in Windows 7 which Microsoft addressed (after a fair dose of community effort), but what you might not know is that there is another and more serious “exploitative” UAC vulnerability breaking exactly what Mark described.



Win7Elevate proof-of-concept -- Running without elevation.

Win7Elevate proof-of-concept v2 (08/Feb/2009).
© 2009 Leo Davidson, all rights reserved. Malicious use is expressly prohibited.
This program attempts to bypass Windows 7's default UAC settings to run the specified command with silent elevation.

Command: C:\Windows\system32\cmd.exe

Arguments:

Directory: C:\Windows\System32

For injection, pick any unelevated Microsoft process with ASLR on:

```
ApntEx.exe  (3140)
Apoint.exe  (2624)
BTTray.exe  (2788)
conhost.exe  (3156)
dllhost.exe  (3352)
dwm.exe  (2356)
explorer.exe  (2464)
firefox.exe  (3968)
hidfind.exe  (3132)
ipoint.exe  (2644)
LocationNotifications.exe  (3280)
reader_sl.exe  (2632)
```

Inject file copy & elevate command
Change a protected folder via the selected process; use the change to launch the command elevated.

Run command directly, with elevation
Run the command normally (without code injection), requesting elevation.

Run command directly
Run the command normally (without code injection), without requesting elevation.

Dear Microsoft: Either local process elevation is important or it isn't. Pick one. Then apply it fairly to *all* applications.

http://www.pretentiousname.com/misc/win7_uac_whitelist2.html

The other UAC exploit, discovered, demoed, extensively documented by Leo Davidson, is a code-injection vulnerability made possible by the new Windows 7 auto-elevation system. To summarize War and Peace into a short story if you will, it allows applications without UAC prompts (medium-level) to run code or other applications with administrative privileges (high-level), assuming the default security configuration in Windows 7 (don’t notify changes to Windows).

It was my original intentions to not publically address this until Windows 7 has been finalized, giving them an opportunity to fix it, which they have not in RC or later builds, but Mark’s article today tells me they’re doing no such thing.

Knowing the vulnerability, I was of surprised to see the article conclude with a direct reference to this exploit.

> Several people have observed that it’s possible for third-party software running in a PA account with standard user rights to take advantage of auto-elevation to gain administrative rights. For example, the software can use the WriteProcessMemory API to inject code into Explorer and the CreateRemoteThread API to execute that code, a technique called DLL injection. […]

> The follow-up observation is that malware could gain administrative rights using the same techniques. Again, this is true, but as I pointed out earlier, malware can compromise the system via prompted elevations as well. From the perspective of malware, Windows 7's default mode is no more or less secure than the Always Notify mode ("Vista mode"), and malware that assumes administrative rights will still break when run in Windows 7's default mode.

Ultimately Mark dismisses the exploit and that's where he lost me.

Mark points out though, excluding this vulnerability, there are actually other known methods for malware to compromise the system via elevation exploits, a flaw in the UAC design. What he misses though is the fact that the problem is more serious in Windows 7 than in Windows Vista.

How these variations of elevation vulnerabilities work is that they all piggyback on elevated application with COM objects that can be exploited to run functions at elevated privileges. However, in Windows Vista, the applications that can be piggybacked on would have displayed a UAC prompt at one point or another to elevate, whereas in Windows 7, there are known Windows executables that can be launched, silently elevated and piggybacked on.

What's more is that this applies not only to malware but to any application. By that I mean legitimate developers can write applications that take advantage of this code-injection vulnerability to make their applications run in administrative privilege without UAC prompts. Of course, the likelihood of this is low, but not impossible. For example, competing softwares could leverage this to make their software appear "less annoying". If you're having to doubt if an application is following the rules, it would damage the reputation of the whole ecosystem.

Putting the "security barrier" jargon aside, I argue as a direct result of the auto-elevation white-list, the UAC in Windows 7 by default is fundamentally less secure than Windows Vista's default. I recognize that UAC was not designed to be a "security feature" to begin with, but with each new version, an operating shouldn't become less secure and expose more risk to the user.

Granted it is highly unlikely Microsoft is willing to revert Windows 7 to UAC-prompt-hell, what they can and should do is communicate that there is a difference in security between the Windows 7 default UAC setting and the "Always Notify" mode. If users then accept the increased risk, then they should be able to enjoy a less annoying Windows.

Thoughts?

Update: I have a video demonstration of this vulnerability in play at an updated post here. The source code has also been released.

Tweet Like 0

← Older Newer →

91 insightful thoughts

Altrus Edwards

June 11, 2009 at 5:12 am

Ah, but he tests the thing on build 7000, are you sure its not been fixed by 7100?

Rafael Rivera

June 11, 2009 at 5:13 am

FYI: Explorer is not an auto-elevated process. I gave you a list…

Rafael Rivera

June 11, 2009 at 5:15 am

@Altrus: Re-read the linked page, specifically under "Quick Windows 7 RC1 Update". Everything still applies to 7100+.

jjpriest25

June 11, 2009 at 5:16 am

They need to do something…either find a way to make UAC actually more secure, or let users know that it is actually less secure–as it is now, it seems awfully shady.

JoeM

June 11, 2009 at 5:21 am

That is why the Vista way was better. People complained, and this is the problem you will get 🙁

Bill Pytlovany

June 11, 2009 at 5:24 am

As a software developer I wouldn’t think twice of taking advantage of this vulnerability to save my users from having to go through the UAC prompt. You’re absolutely right about competitive advantage.

If “I* would do it, anyone would.

Bill

crankenstein

June 11, 2009 at 5:28 am

that’s WHY you leave it turned on AND set to it’s highest priority… it’s a safety net, and a damn good one too. the UAC prompt is not a big deal! *sheesh!*

Brandon

June 11, 2009 at 6:01 am

Microsoft does not care.

Susan

June 11, 2009 at 6:11 am

I have started a campaign to urge people to zip up their slider… check out

http://www.cafepress.com/windows7

Steve Friedl

June 11, 2009 at 6:29 am

Though UAC is not a security boundary, it most certainly is a security feature.. This distinction matters.

And if one really wants to be secure, s/he should run as a **Standard User** instead of an admin no matter what the UAC slider is set to. Ref: http://unixwiz.net/techtips/win7-limited-user.html

jon

June 11, 2009 at 7:09 am

"FYI: Explorer is not an auto-elevated process. I gave you a list…"

That's pedantic hair splitting; Explorer is able to create auto-elevated COM objects like IFileOperation, which is one of the vectors of this vulnerability.

All you have to do is try making a change to a protected system location in Explorer under Windows 7 to see this for yourself.

Stefan

June 11, 2009 at 7:15 am

Actually, you can with UAC's default behaviour start the Task Manager with administrative rights (select in "process" tab "Show processes of all users" or sth. like that) without a UAC conformation. Furthermore you can lanuch any application in context of the Task Manager with administrative without overall one UAC confirmation. Btw, make sure you select "Start this task with administrative rights" 😉

June 11, 2009 at 7:37 am

Stephen

What you guys don't seem to grasp is that, this exploit works only on a PA account (admin account), it doesn't work for a standard user.. the only thing the exploit gains is that:

When an administrator downloads a program and runs it.. it 'owns' the computer

vs

When an administrator downloads a program and runs it.. he has to first 'accept' the action (no password)..

It seems pretty clear to me that the scenario where an admin has downloaded an executed a program, he would of course click accept to a uac prompt anyway.. so the exploit gains nothing..

And for the case where a standard user downloads and executed a program, they would need to find an admin to give them username and password.. the exploit doesn't work on standard users.. so it gains nothing

What I AM concerned with is how many people don't understand the problem, and like how many people didn't understand Vista (ie, never used it) they'll start spouting tons of sh– anyway, which will eventually get to news websites as.. WINDOWS 7 DOOMED TO EXPLODE BY OOBARVIRUS ATTACK! BUY MAC NAO! MAC MAC!

So if at all possible, they need to fix it JUST for the sake of making you people shut the hell up..

But I get the feeling this isn't possible.. not without causing a ton of problems, and they especially won't change it this late in the 'game'.

June 11, 2009 at 7:38 am

Unfreakinbelievable!

GoodThings2Life

This is why I just turn it off on my own systems, and now I'm just gonna pre-emptively go into Group Policy on my work domains and disable it before my rollout… it obviously serves no benefit other than to annoy the everloving piss out of me as an administrator, and it does no good for the user either since a) they never read it with comprehension in the first place and b) they obviously aren’t properly protected. Afterall, my users don’t have local admin rights to start with, and they’re all XP users anyway, so it just saves me having to explain why they’re getting all these annoying prompts once I start a deployment of 7.

Still, why can’t Microsoft realize that it’s time to get off their collective rump-roast and either fix this or nix the so-called “feature” altogether? I consider it a failed experiment.

June 11, 2009 at 8:03 am

I like vistas uac and would have liked more prompts

Tyler

June 11, 2009 at 8:26 am

@Stephen:

Leo Davidson

Say you’ve got a remote execution exploit — or a buffer overflow exploit affecting something most people consider innocent such as JPEG files or (until recently) PDF documents — which allows an attacker to run arbitrary code within a vulnerable process.

If that process isn’t a high-integrity process then that code will find it difficult (not impossible) to gain admin rights and install a rootkit that can potentially *never* be detected (if it hides itself at a low enough level).

With the default Windows 7 settings that code that has gained medium-integrity access can instantly, and without any indication to or interaction with the user whatsoever, jump up to high-integrity and install whatever it wants.

The default settings are what most home users will use. (While most business users will continue to use locked-down accounts and never use elevation at all, just as they did from NT4 onwards.)

If you don't think that's important then argue all you want for UAC prompts to be turned off entirely as, by that logic (which is perfectly reasonable, although not the view I personally hold) there is no value in the UAC prompts at all.

(I'm not saying disable UAC completely; just make it auto-accept all prompts for admin accounts so that people are not bothered. And do it for *all* software, not just the apps which Microsoft wants to give special treatment.)

June 11, 2009 at 8:27 am

Leo Davidson

Sorry, my 2nd paragraph should've started "**With Vista**, if that process isn't a high-integrity process…"

June 11, 2009 at 8:30 am

jon

"What you guys don't seem to grasp is that, this exploit works only on a PA account (admin account)"

Which is the type of account that 99.99% of home users will run with by default.

June 11, 2009 at 8:39 am

Leo Davidson

…and obviously I'm only talking about the default Windows 7 limited-admin account. (I never hide that on my page either.)

Since Microsoft are ignoring this issue we've all given up hope that it'll be fixed. Now it's about educating people about what the new default UAC mode does so that they can make an informed decision to either turn up the UAC level to what it was in Vista, or turn it down to silent elevation for all applications (depending on which side of the fence they are on), and to realise that the default settings make no sense to anyone (except from a marketing perspective).

Obviously people can set things to work how they did on Vista. Or they can set things to silently elevate every request. Or they can run as a standard user (but if they found Vista annoying then they’ll really hate that). What people may not realise yet is that those are options, and why they might be better off changing from the defaults.

The other important thing is the annoyance level of UAC on Vista was pretty much caused by Microsoft’s own code using UAC in a really bad way, showing you too many prompts for single logical actions (and stupid prompts-about-prompts). Microsoft keep harping on about forcing developers to fix their code to work well with UAC, yet they themselves are the worst failures at that task. Rather than do waht they’re telling everyone else to do they’ve given themselves a backdoor to UAC and made UAC a joke (or more of a joke if you thought it was worthless to begin with) in the process. It’s shameful hypocrisy and punishes the 3rd developers who bothered to use UAC properly for mistakes that were Microsoft’s fault.

June 11, 2009 at 8:47 am

GoodThings2Life

@Leo,

Now, actually, you make a good point… maybe it is more about educating the user. As admins, we have the ability to make it as strict (Always Notify) or as least intrusive (Never Notify) as we feel appropriate, AND we can either set the user as an Admin or User or in a domain environment anywhere in between.

When I think about the levels of options, I’m a bit less annoyed by the defaults. Doesn’t mean I won’t continue with my plan of limited user and no UAC for my environment, but at least I have options.

June 11, 2009 at 8:51 am

Lazlo

Just install a decent HIPs and disable UAC. You’re far better protected and have more control…

June 11, 2009 at 9:10 am

T

my default setting is “Always notify me when” because UAC prompt doesn’t bother me. anyway I’m confuse about something when accessing “Change UAC settting” in” Standard User Account (limited Account)” the setting information slider is differently shown .

Here’s what i see when Changing UAC Setting under a Standard User Account (limited user account).

In Administrator User Account it shows “Default – Notify me only when programs try to make changes”

In Standard User Account (limited User Account) it shows “Always notify me (and do not dim my desktop whe:”

I’m just wondering why those information is different?

June 11, 2009 at 9:24 am

reXor

I don’t care either.. about UAC… or Windows 7 tbh 😛
tis a stiiinking pile

June 11, 2009 at 9:25 am

Leo Davidson

@T: That’s because the wording of the options is misleading.

The default for admin accounts should be called “Notify me when non-Microsoft Windows applications request administrator rights (unless they bypass UAC).”

The Standard User Account default is the same* as the highest (non-default) setting for admin users. (* In terms of when prompts appear.) There is no option for Standard User Accounts to auto-elevate anything (for obvious reasons).

June 11, 2009 at 9:42 am

Jeroen

People who know how to use Windows will take a standard user account to do their daily work and a administrator account to do system maintenance.

People who do NOT know how to use Windows will take a administrator account as their account and use UAC as the final line of defence.

From a security perspective: Microsoft should make the standard user a default for installing Windows and give the user the ability to choose an administrator password during installation. This will probably be the next step. But you'll have to enter a password when you get a UAC prompt, hence more UAC-prompt-hell.

From a usability perspective: Windows XP didn't have UAC and everyone was administrator, Vista had too much UAC prompt hell, it's a fine line in the middle. Though auto-elevated processes should not allow DLL injection or any other way to be tampered with… maybe high-integrity processes for all of those to prevent tampering?

I guess there will be a lot of debating about this…

June 11, 2009 at 12:07 pm

nabe

Those who dismiss this problem really have no understanding of how important a correct UAC really is. Since developers now in Windows 7 can use (and WILL use) this "feature" to bypass UAC, applications will be written as they were for pre-Vista systems, read: no support for standard accounts. Any lazy programmer will just ASSume their applications are working OK (since they will be using an Admin acccount) and don't bother testing them under standard accounts…

which means, standard accounts will remain a pain in the ass to use because of old software that still works like this, and new software that will be written to work like this again.

No “good” standard accounts: IT nightmare.

Not to mention the reduction in security for Admin accounts (rootkits all over again…)

LDMartin1959

June 11, 2009 at 1:43 pm

Microsoft once again takes the head in the sand approach: “As long as we ignore the problem and deny that it's a problem, it's not a problem. And anyone who thinks it is a problem just doesn't understand that it isn't a problem as long as we continue to deny that it's a problem. See, no problem.” That's only one reason why my next computer wont be running Windows.

ms

June 11, 2009 at 1:47 pm

Try reading the help file for UAC:

Security Impact for default setting: “It is usually safe to allow changes to be made to Windows settings without you being notified. However, certain programs that come with Windows can have commands or data passed to them, and malicious software can take advantage of this by using these programs to install files or change settings on your computer. You should always be careful about which programs you allow to run on your computer.”

The real point of UAC is to get developers to fix their programs and the default UAC setting is sufficient for this goal. Everybody should be running anti-malware software on their PCs and by default Windows will warn you if you don't have any running. UAC at its default level stops normal applications from making unintended changes to the system. It is not a substitute for anti-malware software (though I wouldn't be surprised to see security ISVs start treating all software that makes CreateRemoteThread calls into Windows binaries as malware).

Joseph Cooney

June 11, 2009 at 2:27 pm

Thoughts? I agree with Steve Friedl – don’t run as admin, run as a standard user.

mark

June 11, 2009 at 3:01 pm

I think its a reasonable concern, I tend to turn mine off when setting up a new build and installing my usual apps, then slide it up to the top for general use. Its a good tool but not the be all, end all of security. Windows 7 will be a large target because of a large user base, so people should not assume its bullet proof. All OS’s have security flaws the most popular will be attacked the most Win 7 and Vista are way better than XP in this regard and will continue to be.
@Susan
Wow tee-shirts for UAC I hope you make your million on it, no worse than the fighting Banana Slug shirts we sell here in Loma Mar.

Luke

June 11, 2009 at 4:23 pm

If I log into my *nix machine as root… and then run a command, that command can generally do whatever it likes.

What is the difference here?

June 11, 2009 at 4:26 pm

@Luke: The difference being whilst on Windows you’re signing in as an administrative user, you don’t have “root” privileges by default. You have to “sudo”, comparable to UAC.

★Long Zheng

Luke

June 11, 2009 at 4:37 pm

@Long Zheng: No. You clearly can’t make that comparison.

The default user account is “Administrator” which states “Administrators have complete access to the computer and can make any desired changes.”

This is like me logging into my *nix box as root. AND THEN sometimes having to do a “sudo”.

Thankfully I never ever log into my *nix as root. I run as a user as Sudo.

Which would be the same as running Windows as Standard account mode.

So what’s the problem?

Luke

June 11, 2009 at 4:38 pm

*correction: I run as a user and then “sudo”.

★Long Zheng

June 11, 2009 at 4:45 pm

@Luke: When people install Windows, or buy a new PC for that matter, are not presented with any choices to run as a standard user. The default has been and will be in Windows 7 an administrative user.

People who are aware of the benefits of switching to and running in a standard user are not the target here, it’s the larger majority of users who don’t know or care enough about Windows security that are at larger risk. Windows 7’s default security model should aim to protect them better, not less.

It's difficult to compare root/sudo with Windows's administrator and UAC because they're based on slightly different security infrastructures. But the fact is in Vista, running as an administrator and turning UAC on always meant that I had to agree to a prompt to elevate privileges. This is no longer true in 7.

Luke

June 11, 2009 at 5:27 pm

@Long Zheng: So what you are saying is, that this is really a Configuration Problem (on Microsoft's behalf) and not the ghastly security vulnerability you suggest in your post?

If so, I agree with you 110%. Microsoft have screwed up the default config, Standard User should be the default account in Windows.

In which case they can (and should) fix the issue, but making Standard User default.

★Long Zheng

June 11, 2009 at 5:34 pm

@Luke: Well it's a combination of both.

There is a security vulnerability by definition, which is "a weakness in a system which allows an attacker to violate the integrity of that system". This vulnerability is however only exposed by the default security configuration in Windows 7.

It's unlikely Microsoft is either going to return Windows 7 to the "Vista mode" of UAC, or make standard users the default account. I suggest that Microsoft shouldn't just ignore the problem but do the next best thing, and that is to acknowledge and educate the user of the security risk with the new default level of UAC.

June 11, 2009 at 7:16 pm

Leo Davidson

@MS: "The real point of UAC is to get developers to fix their programs and the default UAC setting is sufficient for this goal."

Microsoft's definition of UAC changes depending on who you ask and what they're trying to get you to believe at the time.

http://blogs.msdn.com/uac/

"User Account Control (UAC) is a core security feature in the next release of Windows Vista and Windows Server code name Longhorn."

https://blogs.technet.com/jesper_johansson/archive/2006/06/22/438316.aspx

"Once the OS is released, if you absolutely can't stand a security feature that is designed to protect you, by all means, turn it off"

As for Standard User accounts, they are a distraction and an excuse as far as Windows 7 goes. You might as well say "People should use Linux to be more secure" as it's about as relevant and likely to happen. If Windows 8 (or whatever) actually makes Standard User the default, and (crucially) improves the user experience to one that people might actually put up with (not one which is WORSE than the Vista mode MS had to backtrack from after so many complaints), then the argument will hold water.

More discussion & thoughts on the purpose of UAC (and the fact Microsoft themselves conveniently change the stated reason for it to suit whatever argument they are trying to win at the time) here:

http://www.withinwindows.com/2009/06/10/uac-uac-go-away-come-again-some-other-day/

June 11, 2009 at 9:47 pm

Leo Davidson

The new Windows 7 UAC logo has just been unveiled:

http://www.pretentiousname.com/misc/uac_comedy_tragedy_security_theatre.pı

😀

Inspired by Rafael (Within Windows)’s “million dollar question: If UAC wasn’t designed to ultimately protect us from anything, why does its icon resemble a damn shield?” I started thinking about what it might represent if it wasn’t supposed to be a protective shield similar to the firewall icon etc…

June 11, 2009 at 11:11 pm

anon

Standard user is not simply a feasible solution today since Microsoft screwed up and didn’t make it default since NT days. Too many apps break (more than what Vista broke due to UAC) including *unmaintained ones* which are oldie gems. Isn’t that why UAC was introduced in the first place? UAC is a retroactively fitted security feature put because they couldn’t suddenly change the default account to standard after all these years. Now I remember reading somewhere that the Windows 7 changes were always a compromise for convenience. Besides not all malware would try to use DLL injection so the default setting is certainly more secure than saying set-it-to-Never-notify-to-eliminate-annoying-prompts-since-anyways-it-serves-no-purpose. If you don’t want any compromises, set it to Always prompt. It’s not broken, it’s a choice Microsoft has given people who didn’t find Vista’s choices acceptable or who use their own discretion when operating as administrator.

June 12, 2009 at 1:04 am

Brendon Kozlowski

I can’t believe I’m saying this…but I honestly hope they’d take up something similar to OSX’s keychain. Similar to Windows’ firewall applications where it prompts you once for practically all applications (of a specific version) if you wish to allow it – if you do, it remembers it. You can optionally not remember the choice. Require a password when you allow (ala sudo). If we upgrade, repeat the process.

Considering all of the mini-database files that make up the entire OS for Windows 7, I can’t imagine why one that remembers previous choices for UAC could not be implemented, and then re-raise the default security level of UAC to high – if a choice is even needed (or switch the user to a standard user).

June 12, 2009 at 1:39 am

Stephen

@Leo, your example for code injection is invalid.. again it still needs to happen for a PA account, standard user will still prompt for uac with password…

And then the scenario comes down to these:

Vista + low integrity app:
Vulnerability in app allows remote code execution, app is low priv (like IE) and therefore cannot do anything

7 + low integrity app:
(same)

Vista:
Vulnerability in app allows remote code execution, executing code attempts to get elevation.. app the user was using appears to be requesting admin access.. admin accepts

7
Vulnerability in app allows remote code execution, executing code attempts to get elevation, and uses dll injection to bypass user consent.

You see, every time this attack is on an admin.. and the choices come down to: being prompted and hitting continue.. vs. it just happens anyway.

If you think the PA 'grant this admin action' is any kind of barrier, then you have an invalid view of the world.. theres two scenarios where this will happen:

User explicitely downloads an app, runs it.. the app is bad, the uac 'grant this admin action' wouldn't have saved the user, because of course they will consent to something they downloaded..

Vulnerable app the user uses is exploited and remote code executes.. the uac 'grant this admin action' again wouldn't have saved the user, because they will trust the application they are using wants admin access for something..

Nothing to see here.. move along

June 12, 2009 at 1:57 am

Long Zheng

@Stephen: In your scenarios, you are neglecitng one critical fact. The whole point of UAC is to encourage developers to write more standard user applications for reasons including security. As a result of this, applications which prompt for elevation is a flag to the user something is "out of the norm".

Whilst a lot of these prompts in the past has been caused by poorly written applications, as we move forward, it will become definitive applications that prompt for elevation actually require it. But even if we're not there now, a UAC prompt right now tells the user an application is trying to go beyond the norm, and gives the user a choice to trust the application or not.

When assessing the impact of UAC, it is unfair to assume users blindly click these dialogs, as that assumption alone would defeat almost all modern security systems.

June 12, 2009 at 2:26 am

JeffU

Long,

I just don't understand why you're barking up this tree? Who cares if there are security holes still, there will ALWAYS be security flaws by all software companies.

This whole post is stupid, you're just trying to drum up support for your site.

June 12, 2009 at 2:33 am

Leo Davidson

@Stephen:

Your argument appears to be that users will OK any prompt shown to them.

In that case, what is the point of showing them the prompts?

Whether their account is a standard user or an admin, assuming they are the physical owner/admin of the machine – if they're not this entire conversation is moot — then, by your reasoning, they will click OK or type their password into every prompt they see.

If you want to argue that the entire concepts of consent and elevation are worthless then that's your choice but I don't think this is the place as your argument isn't specific to the code-injection vulnerability.

Either way, you should agree with us that the default UAC settings are wrong because. They show people some prompts which you clearly think are a waste of time in all situations and configurations.

Personally, if a UAC prompt appears out of the blue I will not click Continue on it, no matter what app it came from. I'll cancel it, then try to reproduce it while monitoring what the process behind it is up to. This isn't just me talking, either… A recent NVidia driver update had some change (which I think is gone in the even more recent updates) where it triggered a UAC prompt every time you changed resolution via their control panel. That didn't happen in the past so I cancelled it, then looked into what was going on.

Also, the UAC prompts could be improved to show the user more context and information about the action that is about to be performed. Right now they just show the exe name and publisher and GUID. If the prompts showed you a description[1] of what was about to be run then it would be much harder to spoof those prompts and trick users (who bothered to read the prompts[2]). I may trust Blah.exe but if a UAC prompt attribted to Blah.exe appears saying "Hey, I'm about to format C:\" then I'm not going to click Continue.

[1] As produced by the admin-side code. Obviously the non-admin-side code create a description that was a lie but the OS could ask the admin-side component for the description, show that to the user, then only tell the admin-side component to execute the command if the user consented.

[2] Users who don't read prompts cannot really be helped, can they? They'll click on and install anything. However, part of the reason users don't read the prompts is that Microsoft's own code (far more than any other code) shows too many of them, and shows stupid prompts-about-prompts which drive users crazy to the point that they keep clicking until the damn things go away. That's Microsoft's fault and it still happens with standard user accounts.

If Microsoft had done the right thing then the conversation today, and while Win 7 was in development, should have been about how to improve the prompts and make UAC more secure, more informative and less annoying. Instead the conversation is the annoying one we're having right now where everyone assumes there are only the possibilities that we see today and nothing else could possibly have been done.

June 12, 2009 at 2:34 am

Leo Davidson

@JeffU, That's sarcasm, right? I sure hope so…

If not then I assume you leave your front door unlocked, never lock your car (if any), have no firewall, tell everyone your password and so on. Who cares about those security holes as there's always some other ones people might us instead… Yeah…

June 12, 2009 at 2:37 am

Long Zheng

@JeffU: Thats a very pessimistic way of looking at it. Its true all software isn't perfect, but that doesn't mean you shouldn't strive to do better.

But besides, I’d love to hear more about “drumming support for this site”. I mean my current ad revenue is an abysmal double-digit, since you seem to be an expert on it, I’d love to hear how to drum more 🙂

June 12, 2009 at 2:55 am

Voopie

Yah, sure. I want to be prompted “Not being prompted may be less secure! Do you want to be prompted? Y/N” every time I set up a goddamned computer.

That was sarcasm. The IE8 Accelerators page is bad enough.

Just never prompt me, give me some marginal level of security that marginally raises the bar for malware authors and hopefully makes the malware nice and consistent and easy to detect, and let me get on with my nice, sweet, unprompted life.

(Until I try to move a file somewhere. That I own. That deserves a prompt)

June 12, 2009 at 3:03 am

Jeff

Lazlo said… “Just install a decent HIPs and disable UAC. You’re far better protected and have more control…”

What is a “HIP”?

June 12, 2009 at 3:18 am

anon

Host Intrusion Prevention System (http://en.wikipedia.org/wiki/Intrusion-prevention_system#Host-based)

Jeff

June 12, 2009 at 3:30 am

@Stephen… you said… "@Leo, your example for code injection is invalid.. again it still needs to happen for a PA account, standard user will still prompt for uac with password…"

That's true, but one of the points in all of this is that Administrator accounts are and will be the default most people will use… whether they buy a new PC with Windows 7 or they install it themselves.

Manufacturers could do the best thing (security wise) and create a standard user account, but Microsoft still needs to be educating people and make it clear to those installing Windows 7 as to why they might want to go with a Standard User account instead of an Administrator account for normal use.

Leo Davidson

June 12, 2009 at 4:13 am

@Voopie: "Just never prompt me"

The Win 7 defaults still prompt you for some things, so you're against the default settings like the rest of us, right?

The defaults make no sense and please no-one except ignorant users and marketing departments. (You still see some prompts and yet get even less security from the prompts you see. Lose-Lose. People who don't value the prompts still get hassled by them. People who would care about that extra security if they knew it was gone have lost it. Who is actually in favour of this new default mode?)

people

June 12, 2009 at 4:32 am

@Lazlo
any HIPS is more annoying than the UAC

June 12, 2009 at 9:25 am

Matteo Gazzoni

One face of Microsoft pretends that UAC is a security boundary/feature (the marketing one); the other says that it is not. It is clear to me that Microsoft wants a lot of users to use UAC to force a lot of developers to not use it (and so to write programs for standard users).

June 12, 2009 at 9:40 am

Xepol

My thought is that DLL Injection should in and of itself require elevation.

If the calls in question required elevation prompts, then the whole argument would be moot.

And systems might be a scootch safer.

June 12, 2009 at 11:53 am

Voopie

@Leo, the words you put in my mouth are unwelcome.

Yes, I like the defaults.

Yes, I don’t mind the tradeoff, because now when I see a prompt, I’m more wary of it. And it prompts when I want it to – when IE’s about to install an add-on, or when an installer needs admin permissions, or whatever.

I don’t mind that level of prompting. I’m not in the habit of running untrusted unscanned EXEs, so I don’t think I care about that either.

So I’m a bit disinterested in UAC. I appreciate the convenience of a seatbelt, but must I wear it indoors too?

June 12, 2009 at 11:59 am

Voopie

And Leo, your argument is nasty. Next time, try not comparing anyone who doesn't share your opinion to an "ignorant user" or worse "marketing department". It just makes you look a bit snide.

So yes, by your argument, I like the defaults, and I am obviously either a marketing department, in which case I think I gained weight, or I'm an ignorant user, and thus cannot be taken seriously by anyone. Unlike Mr UAC WAS FANTASTIC AND WE WANT IT BACK THE WAY IT WAS BY DEFAULT.

UAC, someone said somewhere, was about beating developers into writing better software. Mission accomplished; turn the defaults down; move along.

June 12, 2009 at 1:12 pm

Leo Davidson

But Voopie, the only things that will display you UAC prompts with the defaults are those things that don't bother to bypass them. With the Windows 7 defaults, bypassing them is easy.

What is the point of that?

If you like the prompts then you presumably don't want things to be able to bypass them.

If you don't like the prompts then you presumably don't want to see them ever.

You don't get either of those things with the defaults.

I wasn't calling you ignorant before; I was assuming that — based on what you said — you don't like the defaults either. If you do like the defaults then, sorry, but I am surprised and wonder why you would like them if you understand things fully. The defaults combine the worst aspects of both points of view (inconvenience and insecurity). I honestly cannot see why anyone would want that combination if they understood what it meant.

June 12, 2009 at 3:26 pm

Code.Red

I’m with Xepol; I don’t understand why medium-integrity applications can inject code into high-integrity applications? This makes no sense at all and if these commands required elevation to be used in the first place, then the exploit is fixed. Right?

June 12, 2009 at 6:33 pm

Leo Davidson

@Code.Red:

Medium-integrity processes cannot inject into high-integrity ones, but with Windows 7’s UAC changes *they don’t have to*.

1) Explorer.exe runs at medium integrity, meaning any other medium integrity process can inject code into it.

2) Explorer.exe, since it’s signed by the Windows publisher, also has the new magic ability to create and use certain high-integrity COM objects *without triggering a UAC prompt*.

Put 1 & 2 together and you get:

3) *Any* medium-integrity process can create and use certain high-integrity COM objects without triggering a UAC prompt.

Those COM objects include one which lets you copy files to protected folders such as System32 and Program Files.

From there it was easy to find a way to copy a file so that an admin process picked it up and then ran whatever code we wanted.

It’s not known what other COM objects are available for this silent elevation. The one I’ve been using is documented but there may be others which are undocumented, but findable by someone with a debugger and time on their hands, and which let you do more than just copy a file… But as it turns out copying a file to a protected folder is all you really need to do.

Miretank

June 13, 2009 at 12:19 am

The whole thing is scary. It's not only about a security flaw, it is about a MAJOR security flaw. Code injection can be easily done like that… it is just not right.

Though I guess there is no more escape for that IMO. RTM is coming soon so…

WELL-DONE EXPOSÉ OF THE DANGEROUS FLAW!

June 13, 2009 at 12:58 am

Mark Russinovich SOLD OUT THE REST OF US for a fat paycheck at Microsoft.

Mark Russinovich's CORPORATE-BULLSHIT SPIN on the UAC debacle and Microsoft's UNWILLINGNESS TO REWRITE WINDOWS AS A SECURE PRODUCT leaves us all wasting millions of dollars and hours on an UNSTABLE O/S and third-party crap from scammers like Symantec and McAfee (who were just fined $750,000 for credit-card scamming of their "beloved" customers.

LONG ZHENG, KEEP UP THE GOOD WORK!!!!!

Brendon Kozlowski

June 13, 2009 at 1:55 am

Wow…that's a bit much, "Well-done".

June 13, 2009 at 2:46 am

Patrick Jakubowski

This is yet another reason to not run even as a protected administrator (PA), to which this exploit applies. Always run your software as a low-rights user. Thankfully, IE in protected mode runs with just such low rights, so applications downloaded from IE will have to jump through an extra hoop (require prompting) the first time they're run.

June 15, 2009 at 4:06 am

Hassan

Windows 7 Rocks. The rest Sucks.

June 20, 2009 at 9:51 am

Nobody Real

You do realize that WriteProcessMemory and CreateRemoteThread are *NOT* non-privileged API's. They require permissions normal users don't have. I think it's highly deceptive to claim they're unprivileged API's when they're not.

July 24, 2009 at 12:04 pm

Ross

If you're not surfing around sites with questionable content, you really don't need any security at all. I've spent the last year in XP Pro with no firewall, and no virus software and with no problems. Every once in a blue moon I'd get whatever AntiVirus is most popular at the time and do a quick scan only to find that I had no malware or viruses or any other malicious software.

That being said, if someone wants to hack you, they will. They just wont be hacking you from a Microsoft Windows cocoon, most likely. Bringing Windows to a hack fight is like bringing a spoon to a gun fight.

July 24, 2009 at 12:11 pm

Ross

Ya Windows 7 rocks… For a word processor / email client. If you want to do any real work or gaming though, your productivity will suffer. You'll find yourself back in XP to remain competitive.

November 14, 2009 at 6:56 pm

GS1

Every piece of software and every OS has vulnerabilities
If it was coded it can be exploited!

As has been said over & over again. The problem is most computer users are too ill informed and ignorant.
They simply fire up their system, run no AV or Firewall and use IE.
They download and run any executable then they are shocked when they find there system has been compromised/infected.

What is needed is more education and a pro active approach,
I used to run XP everyday as Admin (With FF & No/Script) and i NEVER got an infection/virus/malware/worm etc and i am not a system administrator,
I am simply an advanced user.

But i suppose as i type this someone somewhere is trying to find exploits for Windows 7 and its only a matter of time before its unleashed and causes mayhem.

January 2, 2010 at 10:21 pm

Ghilli

I'm agreeing with GS1 – If you run antivirus software you will be FINE; I also used to run firefox (with the no script add-on) nearly everday, i downloaded anything that I wanted, my hard drive space was my limit, and I never got a serious virus, all I EVER got was some addware, and that is not luck.

BTW my computer ran great for the 4-5 years that i used it, and it still works now; though i am having problems with it – the reason i bought a new comp with win7 (also the need to upgrade played a part)

anyway u want a solution to your win7-UAC problem here it is: get some AV and malware protection and stop the viruses from even getting to the point where they are able to "exploit" anything – and if it's a program that you downloaded chances are that you know what it is; if you don't then its your own fault

To be honest with you when I was google-ing and found this i was looking for some info on why certain programs wouldn't run on win7 because of new security features

Just a question: Can ANY of you HONESTLY say that you have gotten a virus (or other) because of THIS "exploit"?

And just so you know, thanks to your article and video, hackers/programmers who wouldn't have figured it out just did.

Nyerguds

January 27, 2010 at 9:53 pm

Wait, you mean YOU can tell in ANY CASE what exploit a virus has used to get into your system? Nope, you can't. Not like the antivirus gives you the virus' source code. The fact remains that this is yet another way for viruses and malware to totally take over your system.

Ghilli

January 2, 2010 at 11:14 pm

I posted twice because I want this to be separate.

You guys ask for change but you don't give a description of what you want. You sit there at your computer reading an article, by someone you don't know you can trust who based it on the info of another person (who is apparently not a trust-able source any way – a few posts up), typing that you agree that its a problem.

You guys wanted more security > they gave it to you > you get mad because it's "annoying" > they tweak it to give you an option to turn it off > u get pissed because now viruses can turn it off as well (not: "it doesn't work")

You admitted in the beginning of your article that you are not a programmer, which brings me to my question do you have any idea how much time goes into writing a program? do you have any idea how long it takes to fix the bugs in the program? do you know how long it takes to even find these bugs? I'm a programmer, and even as a novice I realize the difficulties that exist in this field of work.

Nyerguds

January 27, 2010 at 9:50 pm

If we were only talking about a program, there wouldn't be a problem. This is the entire OPERATING SYSTEM. They had a good opportunity to implement stuff like this when they made NTFS, and they didn't. What's stopping them from making an NTFS2 for their next Windows and integrate a full user rights file system into it?

And I AM a programmer. This isn't about "fixing bugs" at all, because it's not supposed to be an "new implemented features" that can have bugs at all. This is about ignoring a core requirement of the operating system, by building on a previous one instead of starting by revising some of the core.

amn

June 3, 2010 at 9:27 pm

A solution I have been using since XP days, which, as most know, did not have UAC at all, is to work as a User (as opposed to an Administrator) with the default built-in Administrator account for all the system work (upgrade, application installation etc). Additionally, for all the applications I need to run which do require Administrator role simply for running (let's call them "legacy" applications) and not for installing or maintaining anything, there is the "Run As..." menu option, which then use.

UAC is not needed then and can be completely disabled and/or removed.

The above is what UNiX has been doing all along – root vs non-root, and su/sudo.

UAC is worthless, unless the path from keyboard to elevated privileges is secure. Software that can send event to UAC control panel applet to make changes FOR the user without even asking them, is not part of such secure path. And so on…

June 5, 2010 at 12:20 pm

Ghilli

and really unless you're downloading a lot of programs (from random sites) or visiting a lot of porn sites (lol), the chances of you actually getting a virus (if u have virus protection software like avg, mcaffee, etc…) aren't even that significant. I admit there is always the risk no matter which site you visit, but generally speaking if you're careful of which sites you visit you should be okay. Firefox's NoScript is a good thing to have too along with some type of AV software.

June 6, 2010 at 5:22 am

WeaselSpleen

Ghilli, you don't need to visit porn sites or download warez to be exposed to malware. That's 1990s thinking. Today's malware is written by sophisticated teams of developers, and distributed via a wide range of methods, including:
Salting of mainstream advertising systems with fake ads that distribute malware. Yes, Google, Yahoo, and Bing are all now vectors for malware. No, I'm not kidding.
Direct infection of random IPs via zero-day exploits in various third-party packages. Not just browser bugs, but bugs in Flash, Acrobat, and even more obscure products, like the Backup Exec exploit that allowed a remote user to gain full control of your SERVER.
Sophisticated and highly targeted social engineering attacks against individual companies, and even individual people.

FireFox with NoScript is a great way to avoid automatic driveby downloads, but anyone who thinks installing antivirus software and avoiding the red-light district is enough to keep them safe is just a disaster waiting to happen.

In short, don't click on shit unless you know exactly what it is, where it's from, and why you need to click on it.

June 6, 2010 at 5:47 am

Ghilli

Off topic, but not really: Don't you just love it when people tell you that you're wrong, but then they say something that is almost identical to what you just said?

I know I didnt say it exactly what you said but "unless you're downloading a lot of programs (from random sites)" is really close to "In short, don't click on shit unless you know exactly what it is, where it's from, and why you need to click on it."

In other words don't tell me I'm wrong if I'm not, but thanks for providing some fancy terms to back up what I was saying.

February 13, 2011 at 1:13 pm

ttx

i have a good one for you all leave you computer running windows vista or 7 off for 4 weeks and then turn it back on and it will tell you are running a hacked windows and yes i have the real deal

November 6, 2012 at 2:21 pm

lady fitness cure for cancer found

This page definitely has all of the information and
facts I wanted about this subject and didn't know who to ask.

Comments are closed.

Long Zheng

User experience entrepreneur
Melbourne, Australia

I'm a person and stuff. Mostly person, sometimes stuff. Proud introvert.

I make/made stuff people love to use:
MyPal: unofficial Melbourne myki mobile app, Omny Studio: enterprise podcast hosting, PTVGlass: Melbourne bus, tram & train timetable on Google Glass, Map2Glass: type and send addresses to Google Glass, SoundGecko: text-to-speech web reader, ChevronWP7: Windows Phone community unlocking, MetroTwit: Twitter app for Windows, Speedo Plus: Windows Phone GPS app, Bing Image Archive: browse daily backgrounds and Windows UI Taskforce: crowdsourced bug tracker.

Follow 11.5K followers

- YouTube
- Instagram
- Flickr
- LinkedIn
- Email

Proudly powered by WordPress