

Re: Windows vs Linux Security

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.misc/2003-09/2104.html>

n1pop_at_hotmail.com

Date: 09/16/03

Date: 16 Sep 2003 13:12:03 -0700

macquigg@ece.arizona.edu (Dave) wrote in message
news:<a3b19517.0309151004.58ee821@posting.google.com>...
> 3) *User Isolation. Is the system and all of its users protected
> against anything that can happen in another user's account? Viruses,
> random code, etc.*

I think this depends on your point of view. For Windows releases like 9x and MD, I believe there is nothing built into the OS to prevent any process, including Windows Explorer, from accessing or altering another user's files or even the system files. Obscurity in form of hidden and system attributes only works if the process ignores such files, so even OS files may be altered.

But NT, 2000, and XP ("NT") appear to be multi-user environments with administrator and non-administrator access. The default access is administrator, of course, so the rest of the system and other users can be configured, and best practice has been to use a regular account for everyday work (or play) and use administrator for doing things that need such access. The drawback with XP (home only, I've no experience with Pro) is that the admin user does not have to log in on system start unless a password is set or if multiple users are defined.

Unix, Linux, BSD, and all the variant and similar environments ("Unix") are multiuser environments with administrator and non-administrator accounts. Much like NT, Unix requires user accounts and most require passwords (some exceptions include Lindows), and offer similar access restrictions. And, like NT, users must log into an administrator account to perform administrator functions.

IMHO, this item is solved, but only for NT and successors. There may be third-party applications for 9x and ME that allow for similar controls, but it's not part of the kernel.

> 4) *Applications. Are all of the applications designed so that there
> is no way malicious code can be run without tricking the user to run
> it? Do these applications recognize a request to run code (even if it
> is disguised as something else) and provide the user an easy way to*

> *run it in isolation.*

I believe that some Microsoft products, namely Outlook and Internet Explorer, and the two most exploited applications in any Windows environment. I believe that GUI email and browser programs for Unix also may contain some vulnerabilities. After all, the job of a browser is to display a web page and obey the commands therein, so a malicious bit of code may execute regardless of the OS as long as the code can be interpreted correctly (java, for example). Exploiting browser and mail clients seems more prevalent in Windows primarily because of the market share. It is likely that Unix exploits exist, but the number of deployed systems is so low that the effort to exploit is too great.

Third party applications for both mail and web browsing exist for all systems, and many are not prone to the exploits found in IE and OL/OE.

But since IE is the default web browser for all Windows platforms, and OL/OE is the default mail client, an attacker is guaranteed nearly 100% of the Windows systems he or she attacks will have one or both exploitable programs installed and frequently used.

> *Level 3) ... In the previous thread, I
> got not a single response to my challenge for anyone to show me code
> that could destroy anything or access "read-only" information outside
> my "junk" account on Red Hat 8.*

I think the absence of responses merely indicates that those who know are not telling. One can be seen as both white hat and black hat if they know that some exploit exists. However, since I don't wear a hat <g>, I can say that I believe a rootkit could do the job, but I have no experience with rootkits or their potential.

> *On the other hand,
> Microsoft has demonstrated that it can make an enormous unstable
> system stable. Maybe they can fix the security problems by "brute
> force" and lots of money.*

Depending on which release you're talking about, I agree or disagree. But I think part of it also has to do with your hardware and whatever third party apps you have installed. For example, I have XP running on a 2GHz Celery machine and it will run for months without a problem.

But I recently discovered that Forte Agent (newsreader) caused some sort of problem that resulted in a system crash and reboot. There is no warning, it's as if someone pushes the reset button. If I don't use Agent, I don't have that problem with any of the other software I use. I believe, since this system has shared video memory, that I could avoid the problems by installing a video card with dedicated RAM, or I could install additional system RAM.

By comparison, my ME system, 300MHz Celery, with Agent was very stable and even supported the home LAN until it was replaced (OS only) with

RedHat.

And a Mandrake installation I had (9.0) would freeze solid for no apparent reason, and nothing clueful in the logs. I had gone so far as to send syslog output to the printer with zero cache (so everything printed immediately), and I still got nothing. Reinstalling seemed to clear up whatever the problem was.

So I think stability might be in the eye of the user and in the configuration of the particular machine. I remember once many moons ago when Linux was just out of beta there were posts from people saying their uptime was several hundred days. In a comparison, I had Windows 3.1 and SLS 1.0 running alongside each other, and neither crashed or exhibited any problems for several months. Windows did crash first, eventually, and may have crashed sooner if I actually did anything on it.

> *I believe Linux applications*
> *will be more secure.*

I agree that Linux and BSD may be more secure because of their open source nature. Some, mainly supporters of proprietary solutions, will disagree because there is a perceived loss of profitability and control when the source is freely available. But with regard to security and the knowledge that nothing can be hidden in an openly available source file, I think open source would be favored by many.

> *In fact, I find it*
> *re-assuring to get occasional alerts from Red Hat when one of these is*
> *a security patch which affects my system.*

At first, I was wary of such a system that, touted as better than Windows by some advocates, would have such a large list of patches immediately available. I mean, days after RH 9.0 hit the streets I bought a copy and installed it. Then I checked RHN and found nearly 30 patch files totalling nearly 150 megs.

But I began to accept that Linux is in a constantly developing state with various stable plateaus being reached and distributed. As well, I believe that an error discovered in a compiler may have a snowball effect in every binary that program created. And when I installed XP (early this year, so it's had some time), I spent lots of time downloading updates and patches. So the net effect is that I'm getting fixes for oversights on both systems, and neither is better than the other.

> *Almost always, these are*
> *obscure problems that *could* be exploited, but haven't been. The*
> *people who discover these problems get credit for their work, and that*
> *may be one reason they use their talents for good, not evil.*

Being an old troubleshooter, I have always accepted that I will never be able to find all the bugs in anything without a little help. Viewing or reviewing from only one point of view only allows for one perspective. But if someone looks over your shoulders, as it were, they might see something in a slightly different light. Open source is that thing being examined, and the world is looking over the programmer's proverbial shoulder to see if they see anything. Closed source does have a number of people all looking at the same code, but they're looking from the same relative position and may miss the same thing many times.

Those that discover exploits in binary releases and report their findings to the developer are doing good. Even those that report to the general public aren't bad, but their practice sometimes leaves something to be desired.

*> Are there many more undiscovered holes at the application level? No
> doubt there are. At the user-isolation layer? I don't think so, but
> I am listening carefully for any evidence to the contrary.*

As with any product, user interaction is always going to be the changing variable. If someone clicks on an attachment and accidentally executes a trojan, even if their system is protected, the likelihood of that person executing another exploit is greater than one who has either been trained or bitten. And once an exploit gains a foothold in a computer system, it is much easier for it to find other exploits only accessible from inside the OS environment (versus attacking via the network interface).