Why are Linux and Mac OS X safer?

First, look at the two factors that cause email viruses and worms to propagate: social engineering, and poorly designed software. Social engineering is the art of conning someone into doing something they shouldn't do, or revealing something that should be kept secret. Virus writers use social engineering to convince people to do stupid things, like open attachments that carry viruses and worms. Poorly designed software makes it easier for social engineering to take place, but such software can also subvert the efforts of a knowledgable, security-minded individual or organization. Together, the two factors can turn a single virus incident into a widespread disaster.

Let's look further at social engineering. Windows software is either executable or not, depending on the file extension. So if a file ends with ".exe" or ".scr", it can be run as a program (yes, of course, if you change a text file's extension from ".txt" to ".exe", nothing will happen, because it's not magically an executable; I'm talking about real executable programs). It's easy to run executables in the Windows world, and users who get an email with a subject line like "Check out this wicked screensaver!" and an attachment, too often click on it without thinking first, and bang! we're off to the races and a new worm has taken over their systems.

Even worse, Microsoft's email software is able to infect a user's computer when they do something as innocuous as read an email! Don't believe me? Take a look at Microsoft Security Bulletins MS99-032, MS00-043, MS01-015, MS01-020, MS02-068, or MS03-023, for instance. Notice that's at least one for the last five years. And though Microsoft's latest versions of Outlook blocks most executable attachments by default, it's still possible to override those protections. **>>Editor's note: the above vulnerabilities, while valid, all had patches provided by Microsoft months before they were actively exploited by malicious code. This does not negate the fact that the vulnerabilities existed -- however there are also exploits for some Linux mail clients (Pine, mutt) that will execute code on a system when the user views the message as well. To bypass the security fix provided by Microsoft, the user must now take some specific actions as explained in the link above.>>**

This sort of social engineering, so easy to accomplish in Windows, requires far more steps and far greater effort on the part of the Linux user. Instead of just reading an email (... just reading an email?!?), a Linux user would have to read the email, save the attachment, give the attachment executable permissions, and then run the executable. Even as less sophisticated users begin to migrate to Linux, they may not understand exactly why they can't just execute attachments, but they will still have to go through the steps. As Martha Stewart would say, this is a good thing. Further, due to the strong community around Linux, new users will receive education and encouragement in areas such as email security that are currently lacking in the Windows world, which should help to alleviate any concerns on the part of newbies.

Further, due to the strong separation between normal users and the privileged root user, our Linux user would have to be running as root to really do any damage to the system.

He could damage his /home directory, but that's about it. So the above steps now become the following: read, save, become root, give executable permissions, run. The more steps, the less likely a virus infection becomes, and certainly the less likely a catastrophically spreading virus becomes. And since Linux users are taught from the get-go to never run as root, and since Mac OS X doesn't even allow users to use the root account unless they first enable the option, it's obvious the likelihood of email-driven viruses and worms lessens on those platforms.

Unfortunately, running as root (or Administrator) is common in the Windows world. In fact, Microsoft is still engaging in this risky behavior. Windows XP, supposed Microsoft's most secure desktop operating system, automatically makes the first named user of the system an Administrator, with the power to do anything he wants to the computer. The reasons for this decision boggle the mind. With all the lost money and productivity over the last decade caused by countless Microsoft-borne viruses and worms, you'd think the company could have changed its procedures in this area, but no.

Even if the OS has been set up correctly, with an Administrator account and a non-privileged user account, things are still not copasetic. On a Windows system, programs installed by a non-Administrative user can still add DLLs and other system files that can be run at a level of permission that damages the system itself. Even worse, the collection of files on a Windows system - the operating system, the applications, and the user data - can't be kept apart from each other. Things are intermingled to a degree that makes it unlikely that they will ever be satisfactorily sorted out in any sensibly secure fashion.

The final reason why social engineering is easier in the Windows world is also an illustration of the dangers inherent in any monoculture, whether biological or technological. In the same way that genetic diversity in a population of living creatures is desirable because it reduces the likelihood that an illness - like a virus - will utterly wipe out every animal or plant, diversity in computing environments helps to protect the users of those devices.

Linux runs on many architectures, not just Intel, and there are many versions of Linux, many packaging systems, and many shells. But most obvious to the end user, Linux mail clients and address books are far from standardized. KMail, Mozilla Mail, Evolution, pine, mutt, emacs ... the list goes on. It's simply not like the Windows world, in which Microsoft's email programs - Outlook and Outlook Express - dominate. In the Windows world, a virus writer knows how the monoculture operates, so he can target his virus, secure in the knowledge that millions of systems have the same vulnerability. A virus targeted to a specific vulnerability in Evolution, on the other hand, might affect some people, but not everyone using Linux. The growth of the Microsoft monoculture in computing is a dangerous thing for users of Microsoft products, but also for all computing users, who suffer the consequences of disasters in that environment, such as wasted network resources, dangers to national security, and lost productivity (note: the link is to a 880 kb PDF file).

Now that we've looked at the social engineering side of things, let's examine software design for reasons why Linux (and Mac OS X) is better designed than Microsoft when it comes to email security. Microsoft continually links together its software, often not for technical reasons, but instead for marketing or business development reasons (see the previous link for corroboration). For instance, Outlook Express and Outlook both use the consistently-buggy Internet Explorer to view HTML-based emails. As a result, a hole in IE affects OE. Linux email readers don't indulge in such behavior, with two exceptions: Mozilla Mail uses the Gecko engine that powers Mozilla to view HTML-based email, while KMail relies on the KHTML engine that the Konqueror browser uses. Fortunately, both Mozilla and the KDE Project have excellent records when it comes to security.

Further, the email programs themselves are designed to act in a more secure manner. The default behavior of the email program I prefer - KMail - is to not load external references in messages, such as pictures and Web bugs, and to not display HTML. When an HTML-based email shows up in my Inbox, I see only the HTML code, and a message appears at the top of the email: "This is an HTML message. For security reasons, only the raw HTML code is shown. If you trust the sender of this message then you can activate formatted HTML display for this message by clicking here." But even after I activate the HTML, certain dynamic elements that can be introduced in an HTML-based email - like Java, Javascript, plugins and even the "refresh" META tag - do not display, and cannot even be enabled in KMail.

Finally, if there is an attachment, it does not automatically run ... ever. Instead, I have to click it, and when I do, I get a dialog box offering me three options: "Save As ..." (the default), "Open With ...", and "Cancel". If I have mapped a file type to a specific program - for instance, I have associated PDFs with the PS/PDF Viewer, then "Open With ..." instead says "Open", and if I choose "Open", then the file opens in the PS/PDF Viewer. However, in either case, the dialog box always contains a warning advising the user that attachments can compromise security. This is all good, very good.

For all these reasons, even if a few individuals got infected with a virus due to extremely foolish behavior, it's unlikely the virus would spread to other machines. Unlike Sobig.F, which is the fastest spreading virus ever, a Linux-based Virus would fizzle out quickly. Windows is an inviting petri dish for viruses and worms, while Linux is a hostile environment for such nasties.