

Re: On classifying attacks

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2005-07/0255.html>

From: Derek Martin (code_at_pizzashack.org)

Date: 07/16/05

Date: Sat, 16 Jul 2005 12:40:29 -0400

To: bugtraq@securityfocus.com

On Fri, Jul 15, 2005 at 06:40:42PM -0500, James Longstreet wrote:

>
> *On Jul 14, 2005, at 9:39 PM, Derek Martin wrote:*
>
> > *This kind of attack has a name already: it is a trojan horse.*
> <snip>
> > *But is this a remote exploit?*
>
> *No, it's not an exploit at all. Systems are not vulnerable to it*
> *unless a local user runs an executable.*

It seems to me your statement can't be correct, because this is ALWAYS the case. A local exploit requires that a local user run an executable. A remote exploit requires that a local user run an executable, even if that is accomplished merely by booting the system. All exploits require running code, and code doesn't magically start itself... Running code is required, because it is the very running code which is being exploited.

> *The only thing it exploits is trust of email (or similar vector).*

I think this is also not the case. To exploit essentially means to use. These attacks USE the users' trust of e-mail in order to USE a bug to gain access to USE the system for his own purposes... That certainly seems like an exploit to me.

> *Your example involving BIND is a good example of a true remote exploit.*

Thank you! We agree on this, at least.

> *A local exploit is typically categorized as one that requires permissions on the system to begin with, and is used to gain elevated permissions (such as exploiting a setuid program, or causing root to write files through symlink race conditions).*

SecurityFocus Bugtraq: Re: On classifying attacks

We agree on this too.

> *This leaves one significant class of vulnerabilities, however.*

This is essentially the very point of my short essay.

> *Let's imagine for a moment that there is a buffer overflow in
> libjpeg that allows an attacker to create a malicious JPEG which
> can cause any program using libjpeg to execute arbitrary code.
> This should be classified as a remote vulnerability.*

We disagree here. The vulnerability is neither truly remote nor local, in the normal senses as we have defined them here. It is a different kind of vulnerability altogether. The vulnerability is one to automatically triggering trojan horses.... Just as in the case of the fabled Trojan Horse, there is no vulnerability at all until the local users make a decision to trust something (data in this case, rather than a hollowed out horse-shaped monument) from an outside source. In this case, the trust is given implicitly rather than explicitly. This is no different than if I handed you a disk, told you to run the program on the disk, and you did so — resulting in the destruction of your hard drive. Would you call this a remote vulnerability? Of course not. But the mechanism is exactly the same... except that some of the minor details are different.

The only difference is the medium used to deliver the trojan horse is a network instead of a disk, and it is slightly more automated, because you are prone to automatically view the data out of habit. If I did hand you a disk and tell you to run the program on it, you would probably be a lot more wary of doing so than you would of reading your e-mail, wouldn't you? Especially if you don't know me very well. But if you were dumb enough to do so, would you call this a remote exploit? What if I gave you a disk that had an Excel spreadsheet on it, which contained data designed to take over your system using a bug in excel... Is this a remote exploit? I don't think so. Now I use the same excel spreadsheet, but I send it to you in e-mail instead of giving it to you on a disk. In all cases, I have given the data to you. In all cases, there is no exploit at all, until you, the local user, decides to trust the data, and run broken code against it. The only difference is the specific delivery mechanism, and the fact that the average user implicitly trusts data received in e-mail. Because really, what choice do they have?

But this is still not a remote vulnerability. It is a user trust vulnerability, as you said yourself. And it is a vulnerability (or susceptibility) to trojan horse data. The fact that the data just happens to come in via a network is largely irrelevant. A remote user can, IN NO WAY, effect an exploit against this kind of vulnerability merely by his own action. This exploit can not happen unless you, the local user, do it for him. This is the essential reason why it is not a remote vulnerability.

Re: On classifying attacks

SecurityFocus Bugtraq: Re: On classifying attacks

> *Users should be able to trust that opening a JPEG file will only*
> *cause certain code to run, namely decoding and displaying that*
> *JPEG.*

Absolutely they should. But the fact that they DO trust it when it is not worthy of their trust does not make this a remote vulnerability.

--

Derek D. Martin

<http://www.pizzashack.org/>

GPG Key ID: 0x81CFE75D

-
- application/pgp-signature attachment: stored